

UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior



TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Diseño e Implementación de un Sistema de Control Remoto para Vigilancia en Android

Autor: Víctor Manuel Ramírez Marcos

Tutor: Javier Fernández Muñoz

Madrid, junio 2017

Índice

Agradecimientos	x
Resumen	xi
1. Introducción	1
1.1. Motivación	2
1.2. Objetivos	3
1.3. Estructura del documento.....	4
1.4. Acrónimos	6
1.5. Definiciones	7
2. Estado del Arte	9
2.1. Videovigilancia	9
2.2. Detección de movimiento.....	11
2.3. Reconocimiento facial.....	13
2.4. Soluciones actuales en el mercado	14
2.4.1. Securitas Direct	14
2.4.2. IMSEL Seguridad S.A.....	16
2.4.3. Altatec	20
2.5. Herramientas utilizadas y alternativas.....	21
2.5.1. Arduino	22
2.5.2. Raspberry Pi	26
3. Análisis del Sistema	32
3.1. Definición del Sistema	32
3.2. Definición de los Requisitos	34
3.3. Requisitos de Usuario.....	35
3.3.1. Especificación Requisitos de Capacidad	36
3.3.2. Especificación Requisitos de Restricción.....	37
3.4. Análisis de los Casos de Uso.....	38
3.4.1. Roles en el Sistema	40
3.4.2. Especificaciones de los Casos de Uso	41
3.5. Requisitos de Software.....	51
3.3.1. Especificación de Requisitos Funcionales	53
3.3.2. Especificación de Requisitos No Funcionales.....	61
3.4. Matriz de Trazabilidad	64
4. Diseño del Sistema	67
4.1. Diseño de la Arquitectura.....	67
4.1.1. Cliente	68

4.1.2. Servidor	70
4.2. Diseño de Clases	72
4.2.1. Diagrama de Clases	72
4.2.2. Identificación de Atributos y Métodos	74
4.2.3. Diagramas de Secuencia.....	78
4.3. Diseño Físico de Datos.....	86
4.3.1. Base de Datos	86
4.4. Diseño de Interfaz de Usuario.....	88
4.4.1. Página de Login.....	88
4.4.2. Página de Crear Cuenta	90
4.4.2. Página de Inicio.....	94
4.4.3. Página de Vídeo	96
4.5. Especificación del entorno de Desarrollo.....	98
4.5.1. Hardware	98
4.5.2. Sistemas Operativos	99
4.5.2.1 Windows 10.....	99
4.5.2.2 Raspbian Jessie 8.....	99
4.5.2.3 Marshmallow 6.0.1.....	100
4.5.3 Lenguajes de programación	100
4.5.3.1 Java.....	100
4.5.3.2 C#	100
4.5.3.3 Python	101
4.5.4 Lenguajes de marcas	101
4.5.4.1 XML.....	101
4.5.5. Software de Desarrollo.....	102
4.5.5.1 Android Studio	102
4.5.5.2 Geany	103
4.5.5.3 MySQL.....	103
4.5.5.4 VLC.....	103
4.6. Diseño del Circuito Electrónico del Sistema de Videovigilancia	104
5. Implementación y Plan de Pruebas	105
5.1. Tecnologías Utilizadas	105
5.1.1. Cliente	106
5.1.2. Servidor	106
5.1.3. Hardware	107
5.2. Resultado Final del Sistema	108

5.3. Definición del Alcance de las Pruebas	109
5.4. Entorno de Pruebas.....	109
5.5. Especificación de Pruebas	110
5.5.1. Pruebas Unitarias de Caja Negra.....	110
5.5.2. Pruebas de Sistema.....	114
5.6. Análisis de Consistencia.....	121
6. Gestión del Proyecto	123
6.1. Planificación Temporal	123
6.1.1. Planificación Inicial.....	123
6.1.2. Desarrollo Real del Proyecto.....	125
6.2. Presupuesto	127
6.2.1. Presupuesto Total	127
6.2.2. Desglose del Presupuesto	127
6.2.2.1. Costes directos.....	128
6.2.2.2. Costes Indirectos	130
7.2.3. Resumen de Costes.....	130
7. Marco Regulador y Ético	132
7.1. Android Studio	132
7.2. Geany	132
7.3. Python	133
7.4. MySQL.....	133
7.5. VLC.....	133
8. Conclusiones y Trabajos Futuros	134
8.1. Conclusiones	134
8.2. Conclusiones personales	135
8.3. Trabajos Futuros.....	136
9. Referencias	137
Anexo: Summary.....	141

Índice de Tablas

Tabla 1: Requisito RC-01	36
Tabla 2: Requisito RC-02	36
Tabla 3: Requisito RC-03	37
Tabla 4: Requisito RC-04	37
Tabla 5: Requisito RR-01	37
Tabla 6: Requisito RR-02	37
Tabla 7: Requisito RR-03	38
Tabla 8: Formato tabla especificación casos de uso sistema	38
Tabla 9: Rol administrador	40
Tabla 10: Caso de uso CU-01	41
Tabla 11: Caso de uso CU-02	42
Tabla 12: Caso de uso CU-03	43
Tabla 13: Caso de uso CU-04	44
Tabla 14: Caso de uso CU-05	45
Tabla 15: Caso de uso CU-06	46
Tabla 16: Caso de uso CU-07	47
Tabla 17: Caso de uso CU-08	48
Tabla 18: Caso de uso CU-09	49
Tabla 19: Caso de uso CU-10	50
Tabla 20: Caso de uso CU-11	51
Tabla 21: Formato tabla descripción requisitos software	51
Tabla 22: Requisito RF-01	53
Tabla 23: Requisito RF-02	54
Tabla 24: Requisito RF-03	54
Tabla 25: Requisito RF-04	55
Tabla 26: Requisito RF-05	55
Tabla 27: Requisito RF-06	55
Tabla 28: Requisito RF-07	56
Tabla 29: Requisito RF-08	56
Tabla 30: Requisito RF-09	56
Tabla 31: Requisito RF-10	57
Tabla 32: Requisito RF-11	57
Tabla 33: Requisito RF-12	57
Tabla 34: Requisito RF-13	58
Tabla 35: Requisito RF-14	58
Tabla 36: Requisito RF-15	58
Tabla 37: Requisito RF-16	59
Tabla 38: Requisito RF-17	59
Tabla 39: Requisito RF-18	59
Tabla 40: Requisito RF-19	60
Tabla 41: Requisito RF-20	60
Tabla 42: Requisito RNF-01	61
Tabla 43: Requisito RNF-02	61
Tabla 44: Requisito RNF-03	61
Tabla 45: Requisito RNF-04	62
Tabla 46: Requisito RNF-05	62
Tabla 47: Requisito RNF-06	62

Tabla 48: Requisito RNF-07	63
Tabla 49: Requisito RNF-08	63
Tabla 50: Matriz trazabilidad Requisitos Usuario/Requisitos Software.....	65
Tabla 51: Formato tabla definición Clases	74
Tabla 52: Clase CL-01.....	75
Tabla 53: Clase CL-02.....	76
Tabla 54: Clase CL-03.....	76
Tabla 55: Clase CL-04.....	77
Tabla 56: Clase CL-05.....	77
Tabla 57: Clase CL-06.....	78
Tabla 58: Configuración ordenador 1	98
Tabla 59: Configuración ordenador 2	98
Tabla 60: Configuración dispositivo móvil	99
Tabla 61: Formato tabla definición prueba unitaria caja negra.....	110
Tabla 62: Prueba Unitaria Caja Negra PUCN-01.....	111
Tabla 63: Prueba Unitaria Caja Negra PUCN-02.....	112
Tabla 64: Prueba Unitaria Caja Negra PUCN-03.....	112
Tabla 65: Prueba Unitaria Caja Negra PUCN-04.....	113
Tabla 66: Prueba Unitaria Caja Negra PUCN-05.....	113
Tabla 67: Prueba Unitaria Caja Negra PUCN-06.....	113
Tabla 68: Prueba Unitaria Caja Negra PUCN-07.....	114
Tabla 69: Prueba Unitaria Caja Negra PUCN-08.....	114
Tabla 70: Formato tabla definición prueba sistema	114
Tabla 71: Prueba Sistema PS-01.....	115
Tabla 72: Prueba Sistema PS-02.....	116
Tabla 73: Prueba Sistema PS-03.....	116
Tabla 74: Prueba Sistema PS-04.....	117
Tabla 75: Prueba Sistema PS-05.....	117
Tabla 76: Prueba Sistema PS-06.....	118
Tabla 77: Prueba Sistema PS-07.....	118
Tabla 78: Prueba Sistema PS-08.....	119
Tabla 79: Prueba Sistema PS-09.....	119
Tabla 80: Prueba Sistema PS-10.....	120
Tabla 81: Prueba Sistema PS-11.....	120
Tabla 82: Prueba Sistema PS-12.....	121
Tabla 83: Matriz trazabilidad Requisitos/Pruebas	122
Tabla 84: Costes Personal	128
Tabla 85: Costes material informático	129
Tabla 86: Costes material fungible.....	129
Tabla 87: Costes indirectos	130
Tabla 88: Beneficios	130
Tabla 89: Resumen costes.....	131

Índice de Ilustraciones

Ilustración 1: Cámara videovigilancia CamCloud/CamCloudPro.....	15
Ilustración 2: Cámara videovigilancia Smart Cam Outdoor	16
Ilustración 3: Cámaras IP Seguridad Hogar	17
Ilustración 4: Cámaras Domo Seguridad Hogar	17
Ilustración 5: Cámaras Tubulares Seguridad Hogar	18
Ilustración 6: Cámaras IP Megapíxel	19
Ilustración 7: Aplicación Altatec Dispositivo Móvil	21
Ilustración 8: Arduino UNO	23
Ilustración 9: Esquema Arduino UNO	24
Ilustración 10: Módulo Cámara Arduino	25
Ilustración 11: Módulo Movimiento Arduino	26
Ilustración 12: Raspberry Pi 2.....	27
Ilustración 13: Esquema Raspberry Pi 2.....	27
Ilustración 14: Módulo Cámara Raspberry Pi.....	28
Ilustración 15: Sensor Radar NJR4265 J1	30
Ilustración 16: Sensor Térmico Grid-Eye.....	31
Ilustración 17: Visión general del sistema.....	32
Ilustración 18: Caso de uso CU-01.....	41
Ilustración 19: Caso de uso CU-02.....	42
Ilustración 20: Caso de uso CU-03.....	43
Ilustración 21: Caso de uso CU-04.....	44
Ilustración 22: Caso de uso CU-05.....	45
Ilustración 23: Caso de uso CU-06.....	46
Ilustración 24: Caso de uso CU-07.....	47
Ilustración 25: Caso de uso CU-08.....	48
Ilustración 26: Caso de uso CU-09.....	49
Ilustración 27: Caso de uso CU-10.....	50
Ilustración 28: Caso de uso CU-11.....	51
Ilustración 29: Modelo Cliente-Servidor [11].....	68
Ilustración 30: Cliente	69
Ilustración 31: Servidor	71
Ilustración 32: Diagrama de clases.....	73
Ilustración 33: Diagrama Secuencia CU-01	79
Ilustración 34: Diagrama Secuencia CU-02	80
Ilustración 35: Diagrama Secuencia CU-03	80
Ilustración 36: Diagrama Secuencia CU-04	81
Ilustración 37: Diagrama Secuencia CU-05	81
Ilustración 38: Diagrama Secuencia CU-06	82
Ilustración 39: Diagrama Secuencia CU-07	82
Ilustración 40: Diagrama Secuencia CU-08	83
Ilustración 41: Diagrama Secuencia CU-09	84
Ilustración 42: Diagrama Secuencia CU-10	85
Ilustración 43: Diagrama Secuencia CU-11	86
Ilustración 44: Interfaz Login Sistema Vertical.....	88
Ilustración 45: Interfaz Login Sistema Horizontal 1	89
Ilustración 46: Interfaz Login Sistema Horizontal 2	90
Ilustración 47: Interfaz Crear Cuenta Sistema Vertical 1	91

Ilustración 48: Interfaz Crear Cuenta Sistema Vertical 2	92
Ilustración 49: Interfaz Crear Cuenta Sistema Horizontal 1	93
Ilustración 50: Interfaz Crear Cuenta Sistema Horizontal 2	93
Ilustración 51: Interfaz Inicio Sistema Vertical	94
Ilustración 52: Interfaz Inicio Sistema Horizontal 1	95
Ilustración 53: Interfaz Inicio Sistema Horizontal 2	95
Ilustración 54: Interfaz Video Sistema Vertical	96
Ilustración 55: Interfaz Video Sistema Horizontal 1	97
Ilustración 56: Interfaz Video Sistema Horizontal 2	97
Ilustración 57: Circuito Electrónico Sistema Videovigilancia	104
Ilustración 58: Esquema Sistema Videovigilancia Hardware	107
Ilustración 59: Ejemplo visualización aplicación	108

Agradecimientos

A lo largo de mi trayectoria académica ya no solo durante la Universidad, si no en etapas anteriores, tengo que agradecer a todas aquellas personas que se han cruzado en mi camino y me han aportado algo bueno para ser hoy día lo que soy, y poder conseguir lo que a día de hoy he conseguido.

Indudablemente sin mi familia y mis amigos me hubiera supuesto una dificultad extrema añadida, pues ellos son los que me han animado siempre a continuar en cuanto se ha presentado alguna situación difícil, a ellos son a los que les agradezco que este camino haya sido mucho más ameno.

Mi periodo en la Universidad no siempre ha sido bonito, no siempre ha habido buenos momentos, pero gracias a ella mi familia se ha aumentado y eso es algo que agradeceré de por vida.

Mis agradecimientos son para esas personas, que tanto en los buenos como en los malos momentos han estado ahí y sé que estarán siempre.

Resumen

En la realización del presente trabajo fin de grado, se busca desarrollar una aplicación sobre la plataforma Android de manera que el usuario pueda vigilar su casa o el lugar que haya establecido para vigilar. Se busca la cercanía y sencillez al usuario además de seguridad y protección.

Dicha aplicación permitirá al usuario ver en directo los vídeos de las cámaras instaladas así como activar el sistema de detección de movimiento, para que en caso de detectar cualquier tipo de movimiento el usuario sea avisado por una notificación al dispositivo Android en el que tenga instalada la aplicación.

Para el desarrollo de dicha aplicación se utilizará un modelo Cliente-Servidor en el que el Cliente es el dispositivo Android del usuario, y el Servidor es el encargado de controlar el movimiento de las cámaras y avisar al usuario en caso de que se produzca cualquier tipo de movimiento. El Servidor dispondrá de herramientas para controlar el correcto funcionamiento y sincronía de los usuarios y de las cámaras de video vigilancia así como los avisos por movimiento.

Siendo el objetivo final la implementación de dicha aplicación en la plataforma Android y del Servidor para que sea capaz de atender a todas las peticiones que puedan realizar los diferentes usuarios de la aplicación.

1. Introducción

A día de hoy los dispositivos móviles son los más usados en comparación con los ordenadores, televisiones o videoconsolas [1]. Es por ello que las aplicaciones y herramientas de la actualidad deberían de tener un hueco en estos dispositivos, aplicaciones y herramientas que nos ayuden en nuestro día a día.

La gran mayoría de las tecnologías se basan fundamentalmente en Internet, cuyo acceso es fácil y rápido para cualquier dispositivo que sea capaz de conectarse, cualquier usuario a nivel mundial podría acceder a Internet, siendo el 50% de la población mundial usuario de Internet, y 2/3 partes usuarios de telefonía móvil [2].

Gracias al Internet nuestro día a día es más fácil y cómodo, tanto para información como para las comunicaciones. Puedes recibir noticias del otro extremo del planeta prácticamente al instante o comunicarte con otra persona que se encuentre a miles de kilómetros de distancia.

Este concepto también sirve para el tema de monitorización y vigilancia, ya que puedes controlar más de un sitio y ver lo que se está retransmitiendo desde ese lugar. Gracias al Internet y a la seguridad de las redes de comunicaciones hoy día es posible realizar instalaciones de video vigilancia con control remoto y monitorización, con detección de movimiento o reconocimiento facial entre otras funcionalidades.

Esta tecnología es denominada Videovigilancia IP [3], combinando los circuitos cerrados de televisión con las redes de comunicación IP. Permite que la supervisión pueda ser en persona (en local) o que la supervisión pueda ser remota a través de imágenes (fotos o vídeos) pudiendo llegar a incluir audio.

El avance de las tecnologías y de la seguridad nos permite a día de hoy diseñar e implementar sistemas de videovigilancia para su posterior control de forma remota. Este trabajo fin de grado busca desarrollar una herramienta que sea sencilla, eficaz, precisa y sobre todo cercana de cara al usuario final y que proporcione la seguridad que éste necesita.

1.1. Motivación

En referencia a la introducción, las nuevas tecnologías de la actualidad nos permiten desarrollar e implementar sistemas de videovigilancia cuyo acceso y control se puede realizar de manera remota, es decir sin estar presente físicamente.

El poder realizar tanto la vigilancia como el control de manera remota facilita mucho la tarea así como los costes involucrados en ello se disminuyen, al igual que el trabajo empleado en la vigilancia. Las diferentes herramientas de detección y reconocimiento son otras de las ventajas que la tecnología y el avance que nos permiten llevar a cabo una videovigilancia más eficaz y precisa.

Para la supervisión de forma local no es necesaria la conexión a Internet, ya que basta con una red de área local para poder controlarlo remotamente, el Internet entra en escena cuando quieres controlar o monitorizar sin estar en la misma red, lo cual sigue siendo una gran característica y una gran ventaja y avance.

Para ello se pueden usar cámaras IP con sensores de movimiento, infrarrojos y distintos tipos de reconocimiento como por ejemplo el reconocimiento facial. Estas cámaras y su instalación no suele ser económico, cuyo precio varía en función de las características de la cámara, como por ejemplo si retransmite en streaming, si solo guarda en local para su posterior visualizado, la resolución de la cámara, visión nocturna, etc.

Los sistemas de videovigilancia pueden funcionar en su mayoría de manera similar, se realiza la instalación de cámara/sensores necesarios, se establece un protocolo de grabación/captura ya sea detección de movimiento, reconocimiento facial o franja horaria, y finalmente se procede a realizar el aviso cuando se detecta intrusión (en caso de que la implementación consista en ello).

La principal motivación de este trabajo fin de grado es realizar un sistema de videovigilancia preciso, eficaz y económico que cumpla con las funcionalidades básicas de un sistema de videovigilancia, las funcionalidades descritas anteriormente, para su posterior control y monitorización en plataformas Android. Para ello se usarán herramientas como Geany [4] o Android Studio [5] entre otras.

1.2. Objetivos

El objetivo del presente Trabajo Fin de Grado es el desarrollo de un sistema de videovigilancia para la plataforma de Android, tal y como se ha comentado en los apartados anteriores.

Dicho Trabajo Fin de Grado pretende tanto analizar como diseñar y desarrollar todas aquellas funcionalidades que harán posible un sistema de videovigilancia preciso, eficaz, económico y cercano al usuario además de aportar la seguridad que el usuario requiere. El sistema de videovigilancia se encargará de retransmitir por streaming imágenes de las cámaras del sistema así como avisar al usuario en caso de detectar movimiento.

Otros objetivos secundarios del sistema son tener registrados el momento exacto en el que se ha detectado movimiento así como proporcionar las imágenes correspondientes al evento recibido. El desarrollo estará enfocado en la sencillez y eficacia del sistema de acorde a la cercanía máxima con el usuario final.

1.3. Estructura del documento

En el siguiente apartado se mostrará cómo y de qué forma se ha estructurado el presente documento de acorde a la siguiente clasificación:

- **Introducción:** En este apartado se realiza una descripción breve tanto de los objetivos principales que se busca con el presente Trabajo Fin de Grado como de las distintas motivaciones que llevaron a cabo su desarrollo e implementación. Finalmente se menciona de forma breve la estructura que llevará a cabo el presente documento con los apartados que lo forman y en qué consisten cada uno de ellos de forma breve y concisa.
- **Estado del arte:** En este apartado se realizará un análisis en el campo de trabajo que influye al sistema de videovigilancia que se va a desarrollar. Se valorarán las aplicaciones similares existentes a día de hoy además de las aplicaciones en las que el sistema se va a apoyar que se irán describiendo en profundidad a lo largo del presente documento. Se mencionarán los diferentes métodos para el reconocimiento de movimiento existentes, cuáles se podrían haber usado y cuál o cuáles se han usado finalmente y porqué. Se finalizará el apartado con un resumen de forma breve y concisa que consistirá en qué se usa hoy día para sistemas de videovigilancia.
- **Análisis del sistema:** En este apartado se va a realizar de forma detallada un análisis de las funcionalidades principales y secundarias que se van a desarrollar en el presente Trabajo Fin de Grado, todas las partes de las que consta el sistema en su conjunto así como sus correspondientes requisitos y casos de uso que los definen, indicando de esta manera que las necesidades planteadas se han cumplido correctamente. Finalmente se definirán las interfaces de usuario de las que el sistema está formado.
- **Diseño del sistema:** En este apartado se hará especial hincapié en el diseño de la arquitectura, en el diseño de las clases y en el diseño físico de datos que forman parte del sistema global.

- **Plan de pruebas:** En este apartado se definirá el plan de pruebas que se ha llevado a cabo para el sistema en global, junto a unos análisis que mostrarán el rendimiento del sistema.
- **Implementación:** En este apartado se realizará una recolección de todos aquellos aspectos con mayor importancia para el desarrollo del sistema de videovigilancia en su globalidad, aportando tanto las tecnologías de las que se han hecho uso como del propio esquema del proyecto en sí.
- **Gestión del proyecto:** En este apartado se hablará sobre de qué forma se ha llevado a cabo el proyecto inicialmente y de cómo se ha llevado a cabo el proyecto de manera real finalmente. Se aportará un cálculo del presupuesto total del proyecto así como su desglose de gastos.
- **Marco regulador y ético:** En este apartado se incluirá de forma breve un análisis del marco legal por el que el sistema de videovigilancia podría verse afectado en España y en Europa.
- **Conclusiones y trabajos futuros:** En este apartado se realizarán comentarios acerca de las conclusiones que se han obtenido durante y al finalizar el Trabajo Fin de Grado, indicando posibles mejoras del sistema de videovigilancia y de cómo se puede implementar o mejorar en un futuro.

1.4. Acrónimos

- **ABF**: *Automatic Band Filter*
- **ABLC**: *Automatic Black Level Calibration*
- **ADSL**: *Asymmetric Digital Subscriber Line*
- **ADT**: *Android Development Tools*
- **AEC**: *Automatic Exposition Control*
- **AGC**: *Automatic Gain Control*
- **APK**: *Android Application Package*
- **AVI**: *Audio Video Interleave*
- **AWB**: *Automatic White Balance*
- **CCTV**: *Closed Circuit Television*
- **CIF**: *Common Intermediate Format*
- **DivX**: *Digital Video Express*
- **DVP**: *Digital Video Port*
- **FHD**: *Full High Definition*
- **FIFO**: *First In First Out*
- **FLV**: *Flash Video*
- **GHz**: *Giga Hertzio*
- **GPL**: *General Public License*
- **GPU**: *Graphics Processor Unit*
- **HD**: *High Definition*
- **HD-SDI**: *High Definition Serial Digital Interface*
- **I2C**: *Inter Integrated Circuit*
- **IDE**: *Integrated Development Environment*
- **IP**: *Internet Protocol*
- **ISP**: *In System Programming*
- **MEMS**: *Microelectromechanical Systems*
- **MP3**: *MPEG Layer-3*
- **MPEG**: *Moving Picture Experts Group*
- **PIR**: *Passive Infrared Sensor*
- **RAM**: *Random Access Memory*

- **RAM**: *Real Audio Metadata*
- **RGB**: *Reg Green Blue*
- **SCCB**: *Serial Camera Control Bus*
- **SHA-256**: *Secure Hash Algorithm 256*
- **TCP/IP**: *Transmission Control Protocol / Internet Protocol*
- **TVL**: *Television Lines*
- **VDC**: *Voltage Direct Current*
- **VGA**: *Video Graphics Array*
- **VIP**: *Very Important Person*

1.5. Definiciones

- **3G**: Es la abreviación de tercera generación de transmisión de voz y datos a través de telefonía móvil mediante UMTS (Universal Mobile Telecommunications System).
- **Android Wear**: Es un sistema operativo para smartwatches, relojes con sistema operativo Android.
- **Efecto Doppler**: Aumento o disminución de la frecuencia de una onda sonora cuando la fuente que la produce y la persona que la capta se alejan la una de la otra o se aproximan la una a la otra.
- **Flag**: Se refiere a uno o más bits que se utilizan para almacenar un valor binario o código que tiene asignado un significado.
- **Frame**: Es la traducción al inglés de la palabra española fotograma, y representa cada una de las imágenes instantáneas de un vídeo.
- **Framework**: Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.
- **GNU/Linux**: Es el término empleado para referirse a la combinación del sistema operativo GNU y el núcleo Linux.

- **Gradle**: Es un sistema de compilación que reúne en uno solo las mejores prestaciones de otros sistema de compilación, está basado en JVM (Java Virtual Machine).
- **Herramientas Lint**: Es una herramienta Android que sirve para analizar el código de la aplicación en busca de errores comunes, como recursos no utilizados o problemas de usabilidad entre otros.
- **IntelliJ IDEA**: Es un entorno de desarrollo Java creado por JetBrains con dos distribuciones: Community Edition (código abierto) y Ultimate (comercial).
- **Login**: Es el proceso mediante el cual se controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.
- **OpenCV**: Es una biblioteca de código abierto para C# / C++ para el procesamiento de imágenes y visión computarizada.
- **Plugin**: Es un software que garantiza la visualización del contenido en Internet que no está diseñado para que lo procese el navegador.
- **ProGuard**: Es una herramienta Android que se encarga de optimizar, de reducir y de ofuscar el código de las aplicaciones Android.
- **Scintilla**: Es un componente libre de edición de código fuente, siendo su aplicación más conocida SciTE, un editor de texto libre con muchas características útiles para hacer programas simples.
- **Scroll**: Es el término en inglés que en español se utiliza para denominar desplazar o deslizar.
- **Streaming**: Es la distribución digital de contenido multimedia a través de una red de computadoras, de manera que el usuario utiliza el producto a la vez que se descarga.
- **Tablet**: Es una computadora portátil de mayor tamaño que un teléfono inteligente integrada en una pantalla táctil, con la que se interactúa con los dedos.
- **Trimmer**: Traducido al español sería recortadora.
- **Unix**: Es un sistema operativo portable, multitarea y multiusuario, desarrollado en 1969 por un grupo de empleados del laboratorio Bell de AT&T.

2. Estado del Arte

En el presente apartado se procederá a describir y enfocar el entorno, el contexto que se acerca más al sistema de videovigilancia a desarrollar en este Trabajo Fin de Grado, así como el análisis de las alternativas que hay disponibles para llevar a cabo el desarrollo del sistema.

2.1. Videovigilancia

Gracias a la videovigilancia, podemos observar contenido multimedia en directo (streaming) ya sea de nuestra propia casa o del negocio, desde cualquier parte gracias al control remoto y a la monitorización a través de Internet, ya sea desde un ordenador o desde un dispositivo móvil.

Los sistemas de videovigilancia se componen básicamente de un grabador digital, de un soporte de almacenamiento donde guardar las grabaciones o imágenes (suelen ser discos duros), y las propias cámaras que se usen en el sistema para llevar a cabo la videovigilancia. El grabador digital está conectado a Internet a través de un router lo cual nos permite poder controlar y monitorizar la imagen proveniente de las cámaras instaladas del sistema de videovigilancia sin importar desde donde las observemos. El control se puede realizar de manera local o de manera remota a través de Internet [6].

Un sistema de videovigilancia puede llegar a tener diferentes usos dependiendo de a qué tipo de usuario esté orientado dicho sistema, los cuales los podemos diferenciar en varios grupos de manera global:

- **Dueños de un negocio:** en un negocio la seguridad de éste es muy importante. Un sistema de videovigilancia te da la oportunidad de poder vigilar tanto a los empleados como de corroborar el perfecto funcionamiento del negocio, así como de mantener al cliente contento. Puede dejar constancia en el caso de que pueda ocurrir algún tipo de incidencia, ya sea un robo, un hurto en la caja o cualquier tipo de pelea, siempre puedes recurrir a las grabaciones para esclarecer los hechos o incluso para intentar recuperar en el caso de que en el incidente se haya producido cualquier tipo de pérdida.

- **Trabajadores**: este sector son los trabajadores que pasan mucho tiempo fuera de casa, eso conlleva tener la vivienda más vulnerable a cualquier tipo de incidencia como puede llegar a ser un robo o un asalto. Gracias a un sistema de videovigilancia se podrá controlar y monitorizar en todo momento qué es lo que está ocurriendo en tu vivienda desde cualquier parte, en el caso de disponer de empleados en el hogar comprobar si están realizando sus tareas o una tarea tan cotidiana como comprobar si realmente se apagó el fuego de la vitrocerámica.
- **Personas que viven solas**: este sector está orientado a todas aquellas personas que viven solas y por ejemplo les gusta saber quién está llamando al telefonillo. También un sistema de videovigilancia es muy útil para personas mayores que necesiten de seguridad en su domicilio pero que quieran seguir manteniendo su independencia, cualquier familiar puede comprobar en todo momento la situación de la persona mayor.
- **Guarderías**: día a día la demanda de los padres que dejan a sus hijos en guarderías es mayor, para poder ver a sus hijos en todo momento mientras no están con ellos. En el caso de ocurrir cualquier tipo de problema recurrir a las grabaciones para comprobar el porqué de ese problema y la posible solución de éste.
- **Drones**: los drones nos aportan multitud de funcionalidad y aplicaciones a realizar. En el ámbito de la videovigilancia es una solución moderna bastante escogida, ya que ofrece multitud de posibilidades de acorde a la seguridad. Su uso está muy extendido en el control de fronteras, en la supervisión y protección de grandes espacios por parte de la policía o algún tipo de fuerza de seguridad y sobre todo a nivel militar tanto vigilancia como espionaje. El uso privado de los drones para videovigilancia está cada día más extendido ya que permite la protección y la vigilancia de grandes espacios contando con un amplio campo de visión y con velocidad y versatilidad. Los drones pueden llegar a tomar imágenes de gran calidad ya que están equipados con cámaras de seguridad de resolución FHD 180 con estabilizador de imagen, el cual garantiza realizar imágenes de gran calidad así como neutralizar los movimientos del dron [7].

La videovigilancia es un sector amplio y explotado, con la gran novedad de los drones que ha ampliado el abanico de posibilidades y funcionalidades de la videovigilancia así como su campo de visión. Debido a ello el sistema desarrollado pretende ser además de eficaz y seguro, un sistema de videovigilancia cercano y sobre todo sencillo y económico.

2.2. Detección de movimiento

En los sistemas de videovigilancia, la detección de movimiento es un aspecto primordial, ya que puede ser el pilar del propio sistema. A la hora de llevar a cabo esta tarea se presentan 2 opciones: la detección de movimiento vía software o la detección de movimiento vía entrada de alarma. Dependiendo el caso se usará uno u otro o una mezcla de los dos tipos [8].

En general los grabadores vienen con la configuración de fábrica por defecto establecida en grabación por agenda o grabación en continuo, es decir que el dispositivo según se conecta en el sistema empieza a grabar sin descanso toda entrada de vídeo que recibe, llenando los discos duros y reescribiendo las entradas de vídeo más antiguas siguiendo una estrategia FIFO.

Dependiendo del caso es probable que el sistema esté enfocado a separar eventos y analizarlos independientemente del conjunto total. Por ello si se desarrolla el sistema en modo detección de movimiento el grabador solo atenderá a los cambios que se produzcan en las imágenes ya sea detección de movimiento por vía software o detección de movimiento por vía hardware mediante algún sensor externo como por ejemplo un sensor PIR, esto permite que si nadie circula por delante de la cámara o en su caso el sensor, no se realizará grabación alguna.

Las 2 opciones que presenta un sistema de videovigilancia por detección de movimiento son las siguientes:

- **Detección de movimiento vía software:** dicha implementación puede estar tanto en el grabador como en la propia cámara que captura las imágenes. Éste método funciona mediante los cambios en la imagen en comparación con los frames obtenidos anteriormente, sin implicar cualquier tipo de cableado. Dispone de la posibilidad de regular la sensibilidad con la que se realiza la detección sin embargo éste método tiene una clara desventaja y es que ante un cambio brusco de la luz en la imagen es probable que este evento lo considere como movimiento sin haber entrado ninguna persona en acción.
- **Detección de movimiento vía entrada de alarma:** dicha implementación se encuentra integrada en muchos dispositivos sin embargo en el caso de encontrarnos con una gama más entrante, esta implementación no aparece. Éste método implica realizar cableado entre los sensores de detección y el grabador, siendo esta opción una opción hardware en comparación con el anterior método que es una opción software. Sin embargo éste método es más seguro a la hora de detectar movimiento ya que si se produce un evento que consista en un cambio brusco de la luz en la imagen, no se considera como movimiento por lo tanto el grabador no salta.

En definitiva dependiendo del caso y de la disponibilidad en cuanto a infraestructura se refiere se empleará un método u otro o una mezcla entre ambos métodos de detección de movimiento, para consolidar el sistema de videovigilancia.

En el sistema de videovigilancia desarrollado se ha empleado e implementado la detección de movimiento vía software, minimizando así el coste y el uso de materiales extra así como cualquier tipo de cableado que la parte de detección de movimiento hardware pudiera añadir al sistema de videovigilancia.

2.3. Reconocimiento facial

En el apartado anterior se ha hablado sobre el aspecto de la detección de movimiento en un sistema de videovigilancia, siendo éste muy importante a la hora de filtrar contenido o de obtener eventos concretos que hayan sucedido. Sin embargo la detección de movimiento no es la única opción que se puede implementar en un sistema de videovigilancia, siendo el método de reconocimiento facial otro aspecto muy a tener en cuenta.

Al igual que la detección de movimiento, el reconocimiento facial se utiliza en casos determinados dependiendo del uso del usuario final, así como se puede incluso mezclar con la detección de movimiento en un mismo sistema de videovigilancia, considerándose el reconocimiento facial como el complemento perfecto del sistema de videovigilancia a día de hoy, permitiendo reforzar la seguridad.

El reconocimiento facial sirve desde reconocer a una persona como identificar a ésta para verificar su autorización [9]. El software encargado de llevar a cabo la tarea de reconocimiento facial se encarga de buscar las caras que encuentra en el video en tiempo real en una base de datos creada previamente. Dicha base de datos está formada con imágenes de caras, las cuales se pueden llegar a dividir en diferentes secciones según la funcionalidad del sistema de videovigilancia, como por ejemplo dividir la base de datos en las secciones de control de acceso, personas VIP o incluso una sección con delincuentes y personas más buscadas, o aquellas que tengan restringido una serie de accesos a una serie de establecimientos o lugares como puede ser un estadio de fútbol o una determinada tienda.

Una vez encontrada la cara en vídeo y comparada con la base de datos en tiempo real, dependiendo del resultado obtenido pues se realiza una acción consecuente en función del sistema de videovigilancia, como puede ser una denegación de acceso o la activación de una alarma para su posterior tratamiento por las personas que están al cargo del control para efectuar las medidas correspondientes al evento recibido.

El reconocimiento facial es una funcionalidad no contemplada en el sistema de videovigilancia desarrollado en este Trabajo Fin de Grado. Sin embargo es una funcionalidad que se puede añadir perfectamente, ya que el sistema de videovigilancia

desarrollado no es un sistema cerrado, si no que se trata de un sistema en el que las funcionalidades pueden añadirse y quitarse de forma sencilla.

2.4. Soluciones actuales en el mercado

En este apartado se va a proceder a describir las principales soluciones actuales que existen en el mercado para satisfacer un sistema de videovigilancia cumpliendo así con todas las funcionalidades del sistema, así como del usuario.

Para ello se va a proceder a identificar la entidad, organización o empresa que ofrece dicho servicio así como de la forma en que lo ofrece. En el mercado actual destacan las siguientes organizaciones:

2.4.1. Securitas Direct

Esta empresa nació en 1988 como parte del grupo sueco Securitas AB. Posteriormente, se convirtió en una compañía independiente estando actualmente presente en 14 países: España, Suecia, Noruega, Finlandia, Dinamarca, Bélgica, Holanda, Francia, Portugal, Italia, Brasil, Chile, Perú y Reino Unido. Cuenta con más de 10.000 empleados. A lo largo de estos más de 25 años de historia, Securitas Direct se ha convertido en número uno en sistemas de alarmas. Más de dos millones de personas en todo el mundo confían en su protección [10].

Los servicios que ofrece la empresa a día de hoy son los siguientes [11]:

- **CloudCam**: Securitas Direct ofrece una gama de cámaras de videovigilancia a través de las cuales se puede observar, se puede grabar todo aquello que sucede en el establecimiento donde se haya realizado la instalación del sistema de videovigilancia. Se podrá consultar las imágenes en vivo con calidad HD desde cualquier parte del mundo a través de tu dispositivo. CloudCam es el modelo que ofrece Securitas Direct la cual está fabricada por Samsung exclusivamente para la empresa, siendo una cámara portátil, ligera y compacta pudiendo moverla para situarla donde se necesite en función de las necesidades actuales. Idónea para

supervisar empleados del hogar, bebés, niños pequeños y ancianos o personas dependientes. Presenta las siguientes características:

- Resolución Full HD 1080p
 - Grabación por eventos, manual y reproducción remota
 - Función habla/escucha
 - Detección de movimiento y sonido
-
- **CloudCam Pro**: Este modelo es un dispositivo que incorpora la visión nocturna, idónea para supervisar comercios, locales y oficinas. Es como el modelo anterior pero incorporando la visión nocturna, con las siguientes características:
 - Resolución Full HD 1080p
 - Grabación por eventos, manual, grabación manual y reproducción remota
 - Función habla/escucha
 - Visión nocturna
 - Detección de sonido y movimiento
 - Ángulo de visión 128°



Ilustración 1: Cámara videovigilancia CamCloud/CamCloudPro

- **Smart Cam Outdoor:** Este modelo es una cámara pensada para controlar exteriores, preparada para resistir aire libre y supervisar la actividad en jardines, piscinas o terrazas. Este modelo presenta las siguientes características:
 - Grabación con resolución HD
 - Zoom digital
 - Visión nocturna



Ilustración 2: Cámara videovigilancia Smart Cam Outdoor

2.4.2. IMSEL Seguridad S.A.

IMSEL Seguridad, S.A. una de las empresas de seguridad privada en Barcelona especializada en aportar soluciones profesionales en materia de seguridad desde 1996. Desarrollan proyectos en todos los sectores y tanto a nivel industrial como particular. Aportan soluciones reales tanto a pequeños establecimientos comerciales como a grandes superficies empresariales y a particulares preocupados por su seguridad y la de sus familias [12].

Asesoran a los clientes desde la fase inicial del proyecto, suministro e implantación de sistemas, mantenimiento de los mismos y estudian y confeccionan protocolos de actuación absolutamente personalizados para aquellos clientes que lo soliciten.

Los servicios que ofrece esta empresa son los siguientes:

- **Alarmas para el hogar:** Esta empresa ofrece un amplio catálogo de sistemas de seguridad. IMSEL ofrece soluciones personalizadas acorde a la vivienda, necesidades y la protección. Tienen más de 20 años de experiencia en este sector protegiendo hogares, y siendo una de las empresas líderes en alarmas hogar de alta seguridad.
- **Control accesos:** Esta empresa ofrece una solución inteligente para controlar los cierres y la automatización de accesos de una empresa. Cuentan con tecnología para poder realizar la instalación a medida de los requerimientos necesarios.
- **Sistemas de videovigilancia:** Esta empresa ofrece una amplia oferta de soluciones de CCTV así como cámaras de videovigilancia con un presupuesto a medida adaptado a las necesidades de cada cliente. IMSEL realiza desde instalaciones de cámaras IP, Domo o Tubulares hasta el diseño y el desarrollo de un proyecto de sistema de videovigilancia de gran envergadura.



Ilustración 3: Cámaras IP Seguridad Hogar



Ilustración 4: Cámaras Domo Seguridad Hogar



Ilustración 5: Cámaras Tubulares Seguridad Hogar

En cuanto a la seguridad en la empresa, IMSEL dispone de las siguientes tecnologías y aplicaciones de las cámaras de videovigilancia [13]:

- **Sistemas analógicos:** Se trata de la tecnología más antigua, cámaras de videovigilancia analógicas conectadas a un grabador con más o menos entradas de video, en función de la necesidad de cada aplicación. Hoy en día se dispone de equipos analógicos con resoluciones de hasta 1000 TVL (líneas de TV) que permitirán un buen apoyo en caso de tener que confirmar una alarma por intrusión o para aquellas aplicaciones donde no sea necesario un análisis posterior con mucho detalle de una grabación. Permite el acceso desde teléfono 3G, tablet, portátil, etc... para supervisión en remoto de la aplicación, conectando el grabador a una línea ADSL, tanto con IP estática como dinámica.
- **Tecnología HD-SDI:** Esta tecnología también conocida como “HD coaxial” o “megapíxel por coaxial”. Permite obtener imágenes de alta resolución Full HD (1920 x 1080) de hasta 2 megapíxel y transmitir las por cable coaxial, en tiempo real. Esta tecnología está especialmente indicada para aquellos clientes que sobre el cableado de un sistema de circuito cerrado de televisión analógico (convencional) existente, pretenden instalar un sistema que les permita aumentar muy considerablemente la resolución y calidad de las imágenes obtenidas, es decir, para aquellas aplicaciones que pretendan dar un salto cualitativo importante, con una inversión contenida ya que nos permite aprovechar todo el cableado existente.

- **Tecnología IP megapíxel:** Actualmente sin lugar a dudas, es la tecnología que permite ofrecer la herramienta más completa y potente para cualquier tipo de aplicación, sea industrial o particular, debido a su mejor calidad y precisión de imagen y al software de gestión que permitirá un análisis forense de alta calidad. Esta tecnología presenta las siguientes características:
 - Transmisión de vídeo en HD y visualización de grabaciones con un consumo bajo de ancho de banda.
 - Grabación de vídeo en HD sin la necesidad de un disco duro de gran capacidad.
 - Búsqueda de vídeo en HD con mayor rapidez.
 - Reproducir vídeo en HD con la máxima calidad de imagen y el mayor detalle.
 - Exportar vídeos en HD sin pérdida de calidad.
 - Integrar las cámaras de videovigilancia analógicas existentes, convirtiéndolas automáticamente en una parte integral de una infraestructura IP, es decir, actualizar los sistemas de circuito cerrado de televisión analógicos a una red de video IP de alto rendimiento.
 - Migrar los sistemas convencionales existentes a una red de video IP de alto rendimiento de acuerdo a la planificación que más le convenga.



Ilustración 6: Cámaras IP Megapíxel

2.4.3. Altatec

Altatec Seguridad es una empresa de videovigilancia que diseñan nuevas soluciones de videovigilancia que se ajustan de forma eficaz a cualquier situación que se necesite. Son una empresa de seguridad con base tecnológica y homologada por el Ministerio del Interior, instalan y mantienen instalaciones de seguridad electrónica con o sin conexión a Central Receptora de Alarmas. Aplican las últimas novedades tecnológicas en sistemas de videovigilancia, alarma y control de accesos. Trabajan con las marcas líderes en el mercado de la seguridad, asegurando garantía de fabricante. Tienen una solución adaptada a cada tipo de cliente: Administración Pública, gran empresa, pymes, comunidades de vecinos, residencial... [14].

Los servicios que ofrece Altatec para la videovigilancia son los siguientes [15]:

- Cámaras de seguridad y sistemas de grabación de alta calidad dotados de tecnología Megapíxel de máxima resolución gráfica.
- Visualización de imágenes desde cualquier ubicación con fluidez, nitidez y sin cortes.
- Sistema de alarma con capacidad de detección de intrusos, videoverificación en tiempo real y aviso a la Policía.
- Compatibilidad con cualquier dispositivo (móviles, tablets).
- Control de accesos mediante últimas tecnologías (huella digital, tarjetas de proximidad) integrado con el sistema de alarma y la videovigilancia.
- Diseño y desarrollo de proyectos especiales (grandes instalaciones, soluciones sin necesidad de alimentación eléctrica, comunicaciones vía satélite, cámaras de seguridad para empresas...)
- Homologación con la Dirección General de Policía y el Ministerio del Interior (nº 3489).



Ilustración 7: Aplicación Altatec Dispositivo Móvil

2.5. Herramientas utilizadas y alternativas.

Como se ha visto en los apartados anteriores, un sistema de videovigilancia consta de varias partes básicas:

- **Capturador de imágenes y/o vídeo:** Será el encargado de capturar las imágenes o el vídeo del lugar donde esté instalado.
- **Soporte de almacenamiento:** Será el encargado de almacenar las imágenes y vídeo proveniente del capturador.
- **Configuración de grabación:** Será el encargado de determinar en que situación o ante que evento se dispone a activar el capturador, ya sea por franja horaria, por detección de movimiento, reconocimiento facial, de manera continua, etc.

Dichos sistemas están formados por cámaras de seguridad, cámaras IP, de visión nocturna, etc. Todo esto necesita un control y una supervisión en persona o mediante herramientas auxiliares como puede ser una aplicación si se trata de un dispositivo móvil.

Sin embargo estas herramientas no son las únicas con las que poder llevar a cabo un sistema completo de videovigilancia, ya que podemos optar por las siguientes herramientas:

2.5.1. Arduino

Es una plataforma de prototipos electrónica de código abierto) basada en hardware y software, flexibles y fáciles de usar. Está pensado para artistas, diseñadores, como hobby y para cualquiera interesado en crear objetos o entornos interactivos. Arduino puede “sentir” el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos.

El microcontrolador de la placa se programa usando el Arduino Programming Language y el Arduino Development Environment. Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador. Las placas se pueden ensamblar a mano o encargarlas preensambladas; el software se puede descargar gratuitamente.

Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia código abierto, por lo que eres libre de adaptarlas a tus necesidades. Arduino recibió una mención honorífica en la sección Digital Communities del Ars Electronica Prix en 2006. Las características que presenta Arduino son las siguientes [16]:

- **Barato**: Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino preensamblados cuestan menos de 50\$ (45€).
- **Multiplataforma**: El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.

- **Entorno de programación simple y claro:** El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.
- **Código abierto y software extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR-C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.
- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender cómo funciona y ahorrar dinero.



Ilustración 8: Arduino UNO

Su correspondiente esquema detallando los componentes por los que está formado, es el que se puede observar en la siguiente imagen:

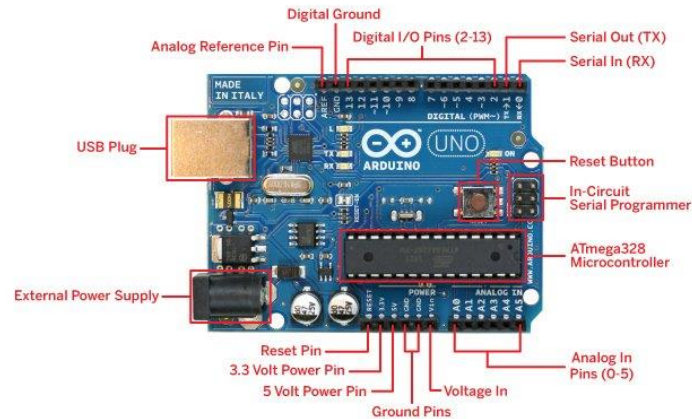


Ilustración 9: Esquema Arduino UNO

Con estas características se convierte en una herramienta útil para convertirse en el dispositivo de recepción de las imágenes acoplando un módulo de cámara y/o un módulo de detección de movimiento, siendo estos de un coste barato. Al ser programable Arduino se puede configurar para que empiece a capturar y/o avisar de movimiento ante el evento que se quiera, ya sea mediante franja horaria, cuando se detecte movimiento o de forma manual. Los principales componentes que formarían un sistema de videovigilancia con Arduino son los siguientes:

- **Cámara VGA**: Este módulo sería el encargado de capturar imágenes y vídeo, cuyas características son las siguientes [17]:
 - Hasta 30 FPS, con sensor de imagen OmniVision
 - Alta sensibilidad adecuada para aplicaciones de baja iluminación.
 - Interfaz SCCB estándar compatible con interfaz I2C.
 - RawRGB, RGB (GRB4: 2:2, RGB565/555/444), YUV (4:2:2) y YCbCr (4:2:2) formato de salida
 - Soporta VGA, CIF, y de una variedad de tamaños CIF a 40×30.
 - Las funciones de control incluyen: control automático de exposición, control automático de ganancia, balance de blancos automático, automático eliminar franjas de luz, controles de calidad de la imagen de

calibración automático del nivel de negro incluyendo la saturación de color, tono, gamma, nitidez ANTI_BLOOM.

- ISP tiene una función de compensación para eliminar el ruido y los pixeles muertos
- Ajuste automático de reducción de ruido.



Ilustración 10: Módulo Cámara Arduino

- **Sensor movimiento PIR:** Este módulo se encargará de la parte de detección de movimiento en el sistema de videovigilancia, cuyas características son las siguientes [18]:
 - Sensor piroeléctrico (Pasivo) infrarrojo (También llamado PIR)
 - El módulo incluye el sensor, lente, controlador PIR BISS0001, regulador y todos los componentes de apoyo para una fácil utilización
 - Rango de detección: 3 m a 7 m, ajustable mediante trimmer (Sx)
 - Lente fresnel de 19 zonas, ángulo $< 100^\circ$
 - Salida activa alta a 3.3 V
 - Tiempo en estado activo de la salida configurable mediante trimmer (Tx)
 - Redisparo configurable mediante jumper de soldadura
 - Consumo de corriente en reposo: $< 50 \mu\text{A}$
 - Voltaje de alimentación: 4.5 VDC a 20 VDC



Ilustración 11: Módulo Movimiento Arduino

2.5.2. Raspberry Pi

Los Mini PCs son siempre una buena opción para disfrutar de toda la potencia de un ordenador pero recurriendo a un tamaño compacto. Los podemos utilizar como servidor de contenidos, conectados al televisor y, por supuesto, como ordenador al uso. Raspberry Pi es uno de los productos más populares para estos fines, tanto por su atractivo precio como por las enormes opciones que trae consigo.

En 2012 llamó la atención de decenas de miles de entusiastas, tras emerger como un ordenador de bajo coste que pudiese llegar al mayor número de usuarios posible, y gracias a la amplia comunidad que aporta valor a este proyecto, existen usos tan diversos como sencillos de implementar con unos pocos conocimientos.

Pero, ¿qué es Raspberry Pi? En realidad, se trata de una diminuta placa base de 85 x 54 milímetros en el que se aloja un chip Broadcom BCM2835 con procesador ARM hasta a 1 GHz de velocidad, GPU VideoCore IV y hasta 512 Mbytes de memoria RAM. En cuanto a su precio, suele estar por debajo de los 40 €, una de las razones que explica su popularidad. De hecho, a finales de 2013 se superaron ya las dos millones de unidades vendidas en todo el mundo [19].



Ilustración 12: Raspberry Pi 2

Su correspondiente esquema detallando los componentes por los que está formado, es el que se puede observar en la siguiente imagen:

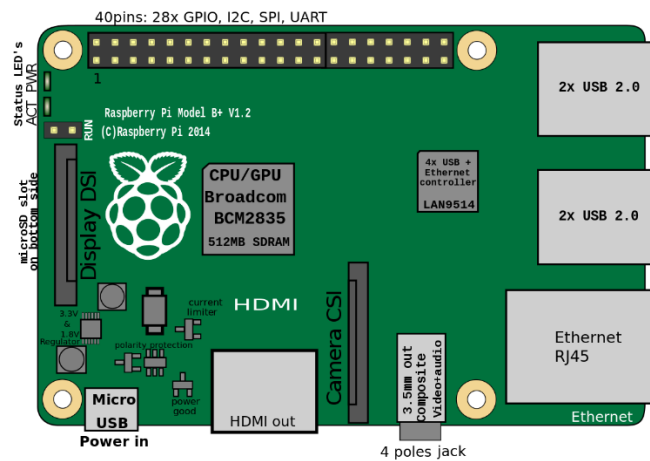


Ilustración 13: Esquema Raspberry Pi 2

Es una herramienta similar a Arduino pero es más potente, de igual manera se le pueden acoplar módulos de cámaras o sensores de movimiento, pero a diferencia de Arduino, Raspberry Pi funciona con sistema operativo, el cual se puede instalar de manera sencilla a través de una tarjeta de memoria SD teniendo a la disposición diversas distribuciones de código abierto y gratuitas.

Los principales componentes que formarían un sistema de videovigilancia con Raspberry Pi son los siguientes:

- **Cámara Raspberry 5MP**: Este módulo sería el encargado de capturar imágenes y vídeo, cuyas características son las siguientes [20]:
 - 1,4 μm X 1,4 μm píxeles con tecnología OmniBSI de alto rendimiento (alta sensibilidad, baja diafonía, ruido bajo)
 - Tamaño óptico de 1/4 “
 - Funciones de control de imagen automáticas:
 - Control automático de exposición (AEC)
 - Balance de blancos automático (AWB)
 - Filtro de banda automático (ABF)
 - Calibración del nivel de negro automático (ABLC)
 - Controles programables para la velocidad de fotogramas, AEC / AGC 16 zonas / posición / control de peso, espejo y lado, recorte, ventanas, y el panorama
 - Puerto de vídeo digital (DVP) Interfaz de salida en paralelo
 - 32 bytes de memoria programable una sola vez incorporada (OTP)



Ilustración 14: Módulo Cámara Raspberry Pi

- **Sensor movimiento PIR**: Este módulo se encargará de la parte de detección de movimiento en el sistema de videovigilancia, cuyas características son similares al módulo descrito anteriormente.

Estas dos herramientas son alternativas para poder desarrollar un sistema de videovigilancia con características, funcionalidades y resultados similares. Sin embargo existen alternativas en cuanto a la detección de movimiento se refiere, ya que se ha planteado un esquema Dispositivo-Cámara-PIR. Existen alternativas al sensor PIR que ofrecen resultados similares, las cuáles son las siguientes [21]:

- **Sensores de radar**: Los sensores de radar pueden detectar movimientos ligeros del cuerpo humano y, si están integrados en un sistema inteligente, podrían resolver los problemas que plantean los sensores PIR. No obstante, para cumplir todos los requisitos, estos sensores deben ser muy pequeños, de coste bajo, potencia baja y con un consumo energético bajo. Los objetos sólidos reflejan la radiación del radar y la radiación reflejada se utiliza para detectar objetos. Una superficie metálica es un objetivo muy bueno para un radar debido a su alta reflexión, e incluso un ser humano refleja de forma bastante satisfactoria gracias al valor ϵ_r alto (valor del agua contenida en el cuerpo). Muchos de los materiales de plástico son más o menos transparentes a la radiación de microondas. Por ello, el módulo sensor puede esconderse con facilidad tras un panel de plástico, lo que representa una ventaja enorme para el diseño exterior del producto final en comparación con los detectores piroeléctricos, que necesitan una lente de Fresnel para funcionar correctamente. El funcionamiento de estos sensores se basa en el efecto Doppler para calcular los datos de velocidad de los objetos a una distancia determinada. La señal reflejada que se irradia hacia el objetivo deseado tiene una frecuencia alterada. Esta variación aporta mediciones directas y muy precisas del componente radial de la velocidad del objetivo en relación con el radar. La diferencia entre la frecuencia reflejada y la frecuencia emitida de una onda para un observador que se mueve en relación con la fuente de la onda se llama efecto Doppler. El radar Doppler se utiliza para detectar objetos en movimiento y evaluar su velocidad. Un objeto reflectante en movimiento dentro del campo del sensor genera una onda senoidal, que es proporcional a la velocidad del objeto. La

medición exacta de la velocidad es una función muy útil, muchos sensores de radar con microondas de RfBeam son capaces de hacerlo incluso en los rangos de velocidades más altas. Sin embargo, para detectar la presencia de seres humanos con tecnología de microondas basta con poder medir una velocidad de 1 m/s (3,6 km/h), pero es necesario poder utilizar un detector muy pequeño, barato y sencillo. Un ejemplo de sensor de radar es el sensor doppler de banda K WaveEye NJR4265 J1, el cual es un sensor de movimiento inteligente diseñado para detectar objetos en movimiento a velocidad baja en distancias cortas, como por ejemplo peatones.

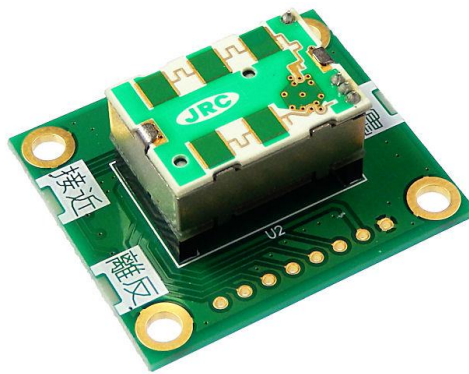


Ilustración 15: Sensor Radar NJR4265 J1

- **Sensor térmico Grid-Eye:** El nuevo dispositivo sensor infrarrojo Grid-Eye de Panasonic es un sensor infrarrojo de tipo termopila, que detecta la cantidad de rayos infrarrojos. Estos sensores de gran precisión, basados en la tecnología avanzada MEMS, cuentan con una detección absoluta de la temperatura, gracias a una superficie bidimensional de 8x8 “píxeles” con una salida digital I2C y un tamaño SMD (de montaje en superficie) miniatura. El dispositivo Grid-EYE es capaz de detectar la temperatura y los gradientes de temperatura sin contacto a lo largo de toda la superficie indicada con un ángulo de visión de 60° gracias a la lente de silicio integrada, ofreciendo una imagen térmica del entorno observado. Diferencia con facilidad objetos múltiples e identifica la dirección del movimiento evaluando la imagen térmica sencilla, al tiempo que no presenta problemas de violación de derechos de privacidad. La detección de cuerpos humanos es posible

gracias a la medición de temperatura sin necesidad de contacto. Los distintos colores de los píxeles representan los gradientes de temperatura.

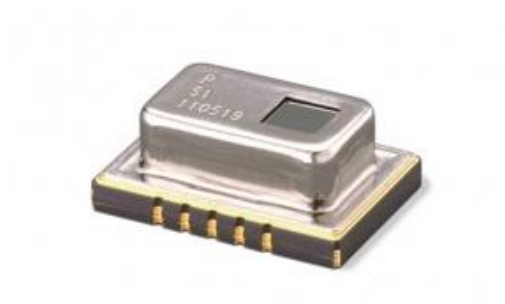


Ilustración 16: Sensor Térmico Grid-Eye

3. Análisis del Sistema

A lo largo de este apartado se va a realizar un análisis del sistema de videovigilancia desarrollado. Para ello dicho análisis se va a basar en la Métrica v.3 [22], la cual consiste en obtener una especificación detallada del sistema que cumpla con las necesidades de los usuarios, sirviendo de punto de partida para realizar más adelante el diseño del sistema.

3.1. Definición del Sistema

Este apartado consiste en realizar una breve descripción del sistema de videovigilancia que se ha propuesto, tanto fijando el alcance que tiene como la identificación propia de los usuarios que forman parte en el sistema.

El sistema que se quiere desarrollar consiste en un sistema de videovigilancia que permita al usuario el control y la monitorización del video así como que lleguen avisos ante eventos como la detección de movimiento la cual realiza la propia cámara.

El sistema propuesto se puede observar en la siguiente ilustración mostrando esta una visión de forma general del sistema:

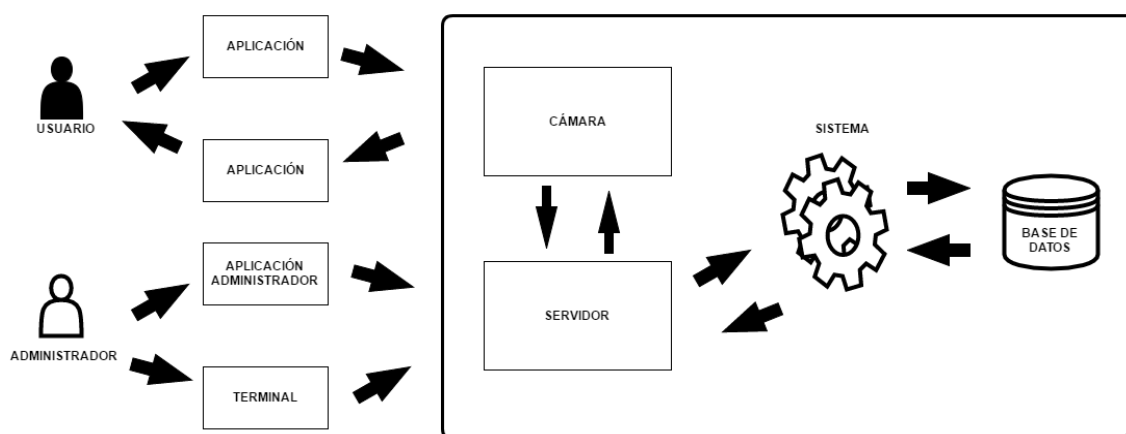


Ilustración 17: Visión general del sistema

Los elementos que forman parte del sistema son los siguientes:

- **Usuario**: Es el usuario que utiliza el sistema de videovigilancia desde la plataforma de Android ya sea para controlar como para monitorizar las cámaras.
- **Administrador**: Es el encargado tanto del correcto funcionamiento como del mantenimiento del sistema. Puede acceder tanto al servidor como a las cámaras para verificar el correcto funcionamiento del sistema.
- **Aplicación**: Es la aplicación implementada para el sistema de videovigilancia con acceso a nivel de usuario, podrá controlar y monitorizar a nivel de usuario su sistema de videovigilancia instalado.
- **Aplicación administrador**: Es la aplicación implementada para el sistema de videovigilancia con acceso a nivel de administrador, podrá controlar y monitorizar a nivel de administrador cualquier sistema de videovigilancia instalado.
- **Terminal**: Interfaz empleada por los administradores para el control, uso, acceso y mantenimiento del servidor y de la aplicación.
- **Cámara**: Son los dispositivos que van a capturar el vídeo o las imágenes, ya sea una o el conjunto de cámaras del sistema de videovigilancia.
- **Servidor**: Es el encargado tanto de recibir las peticiones de los usuarios como de proporcionarles una respuesta a la petición recibida.
- **Sistema**: Es el encargado de que la aplicación funcione correctamente. Esta parte se va a encargar de tratar las peticiones del usuario, clasificarla en función de los requerimientos de la petición, realizar las acciones necesarias y responder a los usuarios con el resultado obtenido, así como las operaciones y consultas necesarias sobre la base de datos.
- **Base de datos**: Es el encargado de almacenar la información necesaria de los usuarios de la aplicación así como de los administradores.

El alcance del sistema hará que los usuarios puedan tanto controlar como monitorizar el sistema de videovigilancia instalado. El sistema se encargará de mostrar en vivo las imágenes captadas por el sistema de videovigilancia así como de notificar si detecta movimiento en caso de que el usuario lo desee. El administrador se encargará del correcto funcionamiento y mantenimiento del sistema para que satisfaga las peticiones de los usuarios.

3.2. Definición de los Requisitos

En este apartado se clasificarán los requisitos del sistema, los cuáles se pueden dividir en dos grandes grupos:

- **Requisitos de Usuario**: Estos requisitos son los encargados de definir los servicios del sistema y de las restricciones bajo las que debe operar.
- **Requisitos de Software**: Estos requisitos son los encargados de realizar una descripción completa del comportamiento del sistema que se va a desarrollar.

Dichos requisitos se dividen en:

- **Requisitos Funcionales**: Estos requisitos son los encargados de definir las funcionalidades que presenta el sistema software o sus componentes, los cuales pueden ser desde cálculos, como detalles técnicos o cómo tratan los datos que reciben entre las funcionalidades que el sistema debe cumplir.
- **Requisitos No Funcionales**: Estos requisitos son los encargados de determinar los criterios que se pueden usar para valorar la acción de un sistema. Son todos aquellos requisitos que se encargan de describir características que no son de funcionamiento, desde el rendimiento hasta la accesibilidad entre otros aspectos.

Los requisitos de software definidos se pueden clasificar a su vez en diversas categorías basándose en la perspectiva del requisito en sí. Dichas clasificaciones son las siguientes:

- Requisitos Funcionales:
 - Sistema
 - Interfaz

- Idioma
- Requisitos No Funcionales:
 - Interfaz
 - Recursos
 - Operatividad

3.3. Requisitos de Usuario

En este apartado se procederá tanto a definir como a clasificar los Requisitos de Usuario del sistema. Para ello se va a emplear una serie de tablas con un formato tabular similar a la tabla que define a los requisitos de software. El formato de la tabla para representar los requisitos de usuario es el siguiente:

Identificador	
Nombre	
Descripción	
Requisitos relacionados	

A continuación se describen los campos que forman la tabla de descripción de requisitos:

- **Identificador**: Este campo representa el código que identifica de manera única al requisito. La nomenclatura que se usará para el nombrado será la siguiente: RX-YY, siendo su significado el siguiente:
 - **R**: Este parámetro indica que se trata de un requisito.
 - **X**: Este parámetro indica a qué grupo pertenece el requisito, puede ser:
 - **C**: Esto indica que el requisito se trata de un Requisito de Capacidad, describiendo la operación que debe de poder realizar el usuario sobre la aplicación o sistema.
 - **R**: Esto indica que el requisito se trata de un Requisito de Restricción, describiendo el cómo el usuario debe de realizar la operación de la aplicación o el sistema.

- **YY:** Este parámetro indica el número de requisito dentro de la clasificación de requisitos.
- **Nombre:** Este campo representa un resumen del requisito formado por un título breve.
- **Descripción:** Este campo representa una descripción completa del requisito junto a todos sus aspectos y características.
- **Requisitos relacionados:** Este campo representa los identificadores de los correspondientes requisitos con los que mantiene algún tipo de relación. Este campo en blanco indica que no presenta relación alguna con algún requisito del sistema.

3.3.1. Especificación Requisitos de Capacidad

Identificador	RC-01
Nombre	Creación cuenta e inicio de sesión
Descripción	El usuario deberá poder crearse una cuenta en la aplicación para poder posteriormente iniciar sesión.
Requisitos relacionados	

Tabla 1: Requisito RC-01

Identificador	RC-02
Nombre	Detección de movimiento
Descripción	El usuario deberá poder activar y desactivar la detección de movimiento en la aplicación.
Requisitos relacionados	RC-01

Tabla 2: Requisito RC-02

Identificador	RC-03
Nombre	Ver vídeo
Descripción	El usuario deberá poder ver el vídeo del sistema de videovigilancia así como dejar de verlo.
Requisitos relacionados	RC-01

Tabla 3: Requisito RC-03

Identificador	RC-04
Nombre	Salida aplicación
Descripción	El usuario deberá poder salir de la aplicación sin cerrar sesión o cerrando sesión.
Requisitos relacionados	RC-01

Tabla 4: Requisito RC-04

3.3.2. Especificación Requisitos de Restricción

Identificador	RR-01
Nombre	Idioma aplicación
Descripción	El usuario deberá poder entender el idioma en el que está desarrollado la aplicación.
Requisitos relacionados	

Tabla 5: Requisito RR-01

Identificador	RR-02
Nombre	Diseño interfaces
Descripción	El usuario deberá poder navegar fácilmente por la aplicación siendo las interfaces sencillas e intuitivas.
Requisitos relacionados	RC-01, RC-02, RC-03, RC-04, RR-01

Tabla 6: Requisito RR-02

Identificador	RR-03
Nombre	Ejecución aplicación
Descripción	El usuario deberá poder ejecutar la aplicación en su dispositivo Android
Requisitos relacionados	

Tabla 7: Requisito RR-03

3.4. Análisis de los Casos de Uso

En este apartado se va a proceder a realizar un análisis en profundidad de las funcionalidades que el sistema pretende llevar a cabo. Con este análisis se busca conseguir de manera global una imagen a alto nivel del sistema destapando todas aquellas necesidades que dicho sistema debe cubrir.

Para llevar a cabo dicho análisis a alto nivel del sistema se realizan una serie de casos de uso los cuales describen las actividades y funciones que debería llevarse a cabo para el correcto funcionamiento de la aplicación del sistema de videovigilancia en cuanto al ámbito de uso de los usuarios se refiere.

Los casos de uso descritos se definirán mediante una serie de tablas con el formato que se describe a continuación:

Identificador		Nombre	
Actores			
Objetivo			
Escenario			
Condiciones de fallo			

Tabla 8: Formato tabla especificación casos de uso sistema

Aparte de la tabla anterior, se añadirá junto al caso de uso una imagen que mostrará el orden que lleva a cabo dicho caso de uso.

Los campos que forman la tabla de los casos de uso son los siguientes:

- **Identificador**: este campo representa el código identificativo único de cada uno de los casos de uso del sistema. Para ello se llevará a cabo la siguiente nomenclatura para nombrarlos: CU-XX, donde XX es el número del caso de uso del sistema que lo identifica.
- **Nombre**: este campo representa el nombre que se le da al caso de uso.
- **Actores**: este campo representa los posibles roles que puede tener el usuario del caso de uso.
- **Objetivo**: este campo representa una explicación de manera breve y concisa del proceso que se lleva a cabo durante el caso de uso del sistema.
- **Escenario**: este campo representa una descripción esquemática de las fases de las que está formado el caso de uso del sistema.
- **Condiciones de fallo**: este campo representa una descripción de los posibles errores que pueden surgir durante el caso de uso del sistema así como las posibles respuestas del sistema para solventar dichos errores.

Antes de proceder a la descripción de los diferentes casos de uso que presenta el sistema, se procederá a definir los diferentes roles que existen dentro del sistema.

3.4.1. Roles en el Sistema

Para que el sistema pueda desempeñar las funcionalidades que se quieren llevar a cabo, el sistema debe considerar los siguientes roles:

Administrador	
Descripción del rol	El administrador del sistema será el encargado de controlar que el sistema de videovigilancia funcione correctamente así como de resolver las posibles incidencias que puedan ocurrir.
Funciones del rol	El administrador del sistema configurará el sistema de videovigilancia, resolverá los distintos tipos de incidencias que puedan ocurrir, se encargará del mantenimiento del sistema así como de su actualización o agregación y eliminación de componentes o funcionalidades.

Tabla 9: Rol administrador

Usuario	
Descripción del rol	El usuario del sistema podrá utilizar la aplicación para el uso del sistema de videovigilancia instalado.
Funciones del rol	El usuario del sistema podrá ver en vivo las imágenes de las cámaras del sistema de videovigilancia, así como decidir cuándo detectar movimiento.

Tabla 3: Rol usuario

3.4.2. Especificaciones de los Casos de Uso

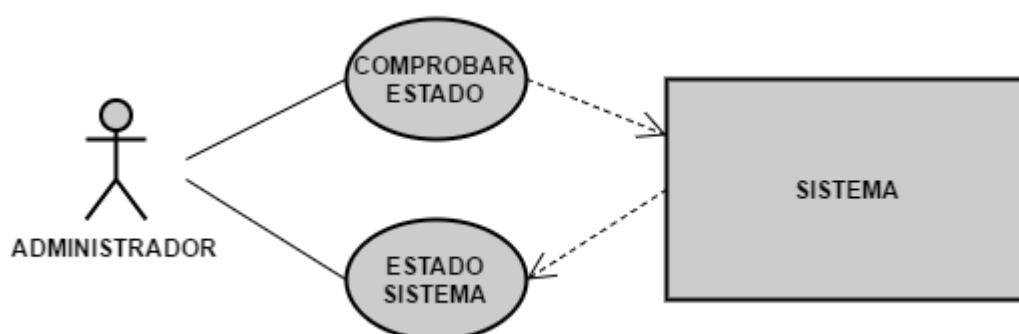


Ilustración 18: Caso de uso CU-01

Identificador	CU-01	Nombre	Mantenimiento
Actores	Administrador del sistema		
Objetivo	El administrador del sistema comprueba el correcto funcionamiento del sistema de videovigilancia, observando la presencia o la no presencia de incidencias y si se han resuelto.		
Escenario	1. El administrador del sistema accede al sistema de videovigilancia. 2. El administrador del sistema comprueba el estado del sistema y registra su estado.		
Condiciones de fallo	<ul style="list-style-type: none"> - Imposibilidad de acceder al sistema. Se reintentará el acceso al sistema y en caso de no poder acceder se contactará con el usuario para comprobar el estado del sistema. - Imposibilidad de obtener el estado del sistema. Se contactará con el usuario para comprobar el estado del sistema. 		

Tabla 10: Caso de uso CU-01

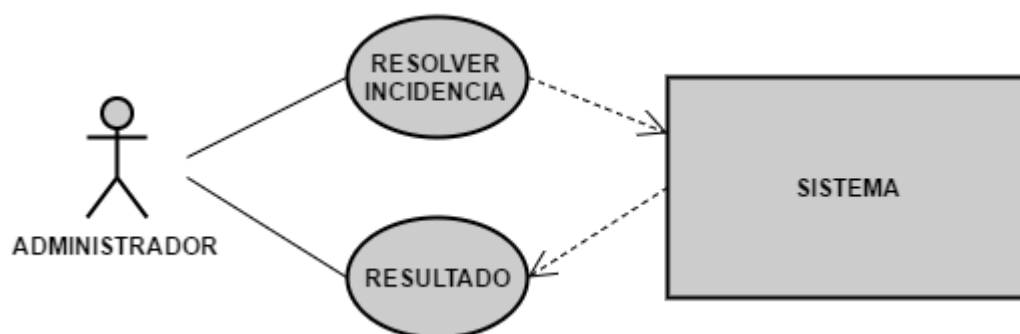


Ilustración 19: Caso de uso CU-02

Identificador	CU-02	Nombre	Incidencia
Actores	Administrador del sistema		
Objetivo	El administrador del sistema resuelve la incidencia detectada al comprobar el estado del sistema o recibida desde éste.		
Escenario	1. El administrador del sistema resuelve la incidencia recibida o detectada. 2. El administrador del sistema registra el estado de la incidencia, ya sea resuelta o pendiente.		
Condiciones de fallo	Imposibilidad de resolver la incidencia. Se registraría la incidencia como pendiente y se probaría otro método de resolución de la incidencia.		

Tabla 11: Caso de uso CU-02

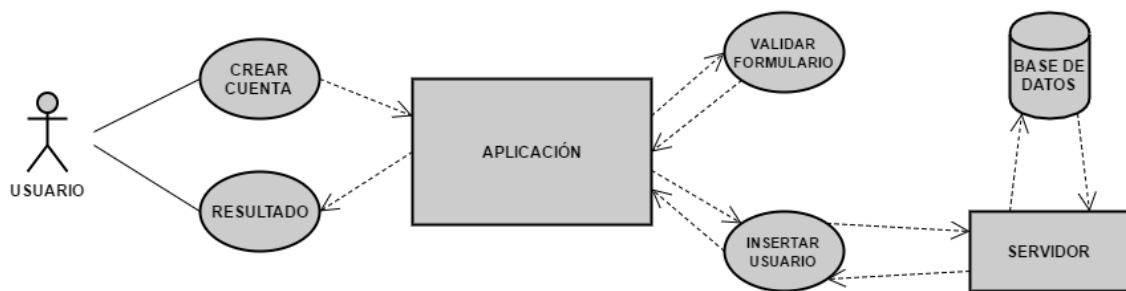


Ilustración 20: Caso de uso CU-03

Identificador	CU-03	Nombre	Crear cuenta
Actores	Usuario		
Objetivo	El usuario crea una cuenta de acceso a la aplicación del sistema de videovigilancia		
Escenario	<ol style="list-style-type: none"> 1. El usuario recibe un código del sistema para rellenar el formulario de registro de usuarios de la aplicación. 2. El usuario rellena todos los campos para crearse la cuenta en la aplicación del sistema. 3. La aplicación valida el formulario de registro. 4. La aplicación inserta el usuario en el sistema. 5. El servidor recibe los datos del usuario y los inserta en la base de datos. 		
Condiciones de fallo	<ul style="list-style-type: none"> - No recepción del código del sistema. Se reenviaría un código nuevo del sistema para crear la cuenta. - Imposibilidad de crear la cuenta. Se muestra un mensaje de error y se recomienda intentar de nuevo más tarde. - Error de formato en los campos de registro. Se muestra un mensaje de error en el campo afectado. - Error al insertar el usuario en el sistema. Se muestra un mensaje de error y se recomienda volver a intentar más tarde. 		

Tabla 12: Caso de uso CU-03

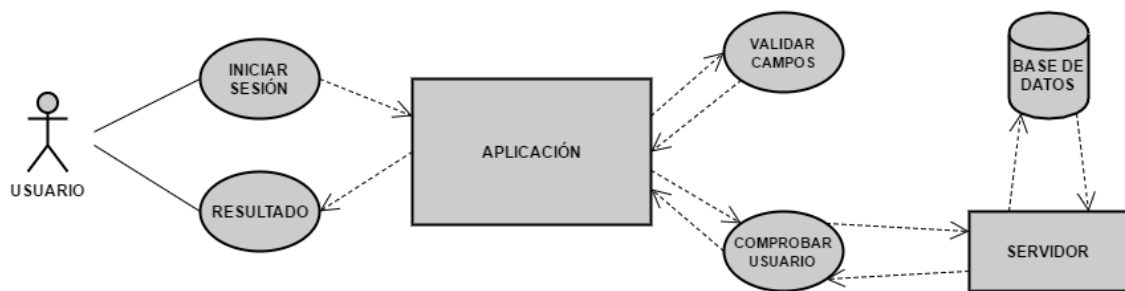


Ilustración 21: Caso de uso CU-04

Identificador	CU-04	Nombre	Iniciar sesión
Actores	Usuario		
Objetivo	El usuario inicia sesión en la aplicación del sistema con una cuenta creada anteriormente.		
Escenario	<ol style="list-style-type: none"> 1. El usuario introduce usuario y contraseña. 2. El usuario envía los datos introducidos. 3. La aplicación valida los campos y manda los datos del usuario al servidor. 4. El servidor recibe los datos del usuario y los comprueba en la base de datos. 		
Condiciones de fallo	<ul style="list-style-type: none"> - Usuario o contraseña incorrecta. Se muestra un mensaje de error de coincidencia de usuario y/o contraseña. - Imposibilidad de validar el usuario con la base de datos. Se muestra un mensaje de error y se recomienda intentarlo más tarde. 		

Tabla 13: Caso de uso CU-04

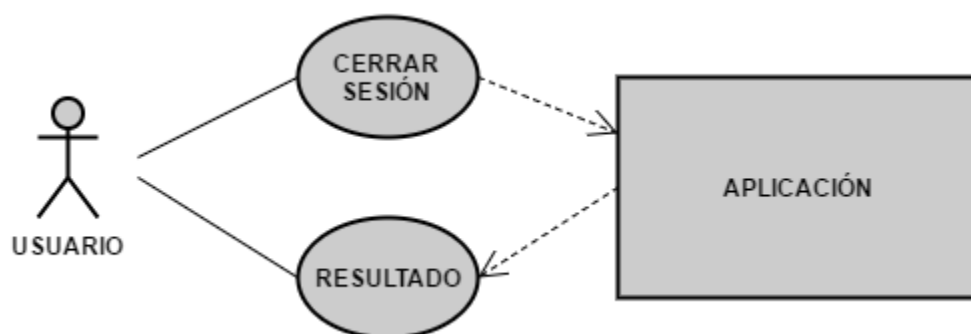


Ilustración 22: Caso de uso CU-05

Identificador	CU-05	Nombre	Cerrar sesión
Actores	Usuario		
Objetivo	El usuario cierra sesión en la aplicación del sistema con la cuenta con la que ha accedido al sistema.		
Escenario	1. El usuario cierra sesión en la aplicación del sistema.		
Condiciones de fallo	Imposibilidad de cerrar sesión. Se muestra un mensaje de error y se recomienda intentarlo más tarde.		

Tabla 14: Caso de uso CU-05

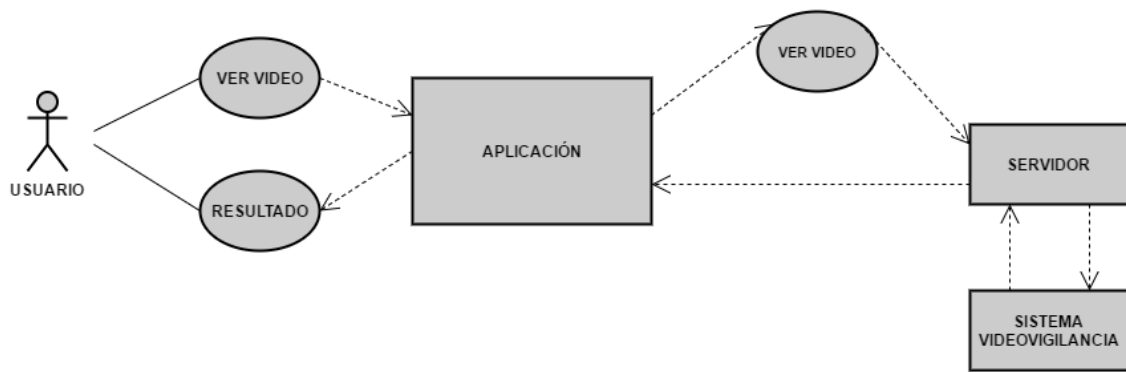


Ilustración 23: Caso de uso CU-06

Identificador	CU-06	Nombre	Ver video
Actores	Usuario		
Objetivo	El usuario ve en vivo el video proveniente del sistema de videovigilancia.		
Escenario	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de ver en vivo en la aplicación del sistema. 2. La aplicación recibe la petición y la envía al servidor. 3. El servidor recoge el video que retransmite el sistema de videovigilancia. 4. El servidor envía el vídeo a la aplicación. 5. La aplicación muestra al usuario el vídeo proveniente del sistema de videovigilancia. 		
Condiciones de fallo	<ul style="list-style-type: none"> - Imposibilidad de recibir la petición. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de conectar con el servidor. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de conectar con el sistema de videovigilancia. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de recibir o de enviar el vídeo. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de mostrar el vídeo al usuario. Se muestra un mensaje de error y se recomienda intentar más tarde. 		

Tabla 15: Caso de uso CU-06

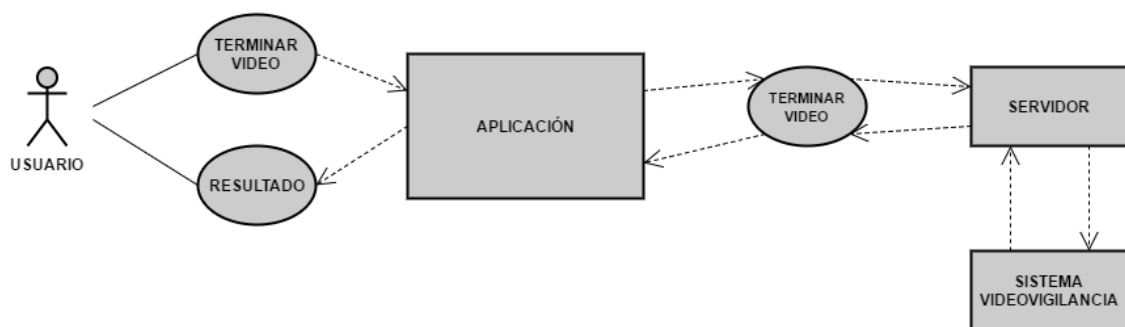


Ilustración 24: Caso de uso CU-07

Identificador	CU-07	Nombre	Terminar video
Actores	Usuario		
Objetivo	El usuario termina de ver el video en vivo proveniente del sistema de videovigilancia		
Escenario	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de terminar de ver en vivo en la aplicación del sistema. 2. La aplicación recibe la petición y la envía al servidor. 3. El servidor para de enviar el video a la aplicación. 4. El servidor para de recibir vídeo proveniente del sistema de videovigilancia. 5. La aplicación deja de mostrar al usuario el vídeo proveniente del sistema de videovigilancia. 		
Condiciones de fallo	<ul style="list-style-type: none"> - Imposibilidad de recibir la petición. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de conectar con el servidor. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de conectar con el sistema de videovigilancia. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de dejar de enviar el vídeo. Se muestra un mensaje de error y se recomienda intentar más tarde. 		

Tabla 16: Caso de uso CU-07

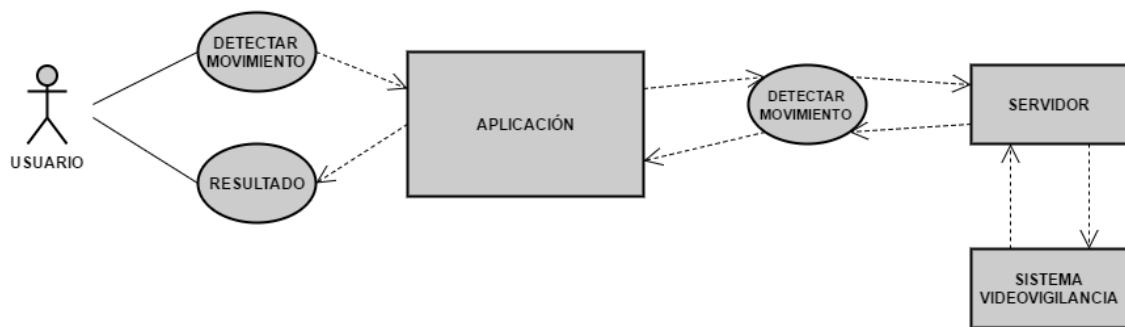


Ilustración 25: Caso de uso CU-08

Identificador	CU-08	Nombre	Activar detección movimiento
Actores	Usuario		
Objetivo	El usuario pulsa el botón de activar la detección movimiento		
Escenario	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de activar la detección de movimiento. 2. La aplicación recibe la petición y la envía al servidor. 3. El servidor recibe la petición y la manda al sistema de videovigilancia. 4. El sistema de videovigilancia recibe la petición y comienza a ejecutar el software de detección de movimiento. 		
Condiciones de fallo	<ul style="list-style-type: none"> - Imposibilidad de recibir la petición. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de conectar con el servidor. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de conectar con el sistema de videovigilancia. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de ejecutar el software de detección de movimiento. Se muestra un mensaje de error y se recomienda intentar más tarde. 		

Tabla 17: Caso de uso CU-08

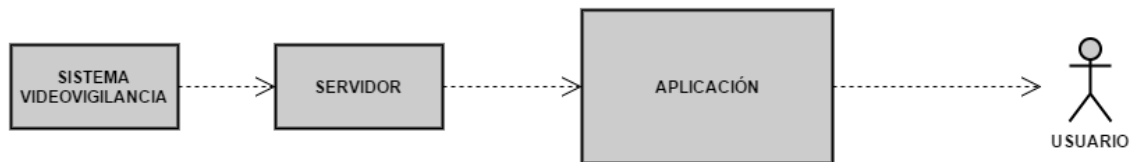


Ilustración 26: Caso de uso CU-09

Identificador	CU-09	Nombre	Movimiento detectado
Actores	Usuario y Sistema		
Objetivo	El sistema de videovigilancia detecta movimiento y manda el aviso al servidor.		
Escenario	<ol style="list-style-type: none"> 1. El sistema de videovigilancia detecta movimiento, registra el evento y manda el aviso al servidor. 2. El servidor recibe el aviso y lo manda a la aplicación. 3. La aplicación recibe el aviso y crea una notificación para el usuario. 		
Condiciones de fallo	<ul style="list-style-type: none"> - Imposibilidad de registrar o mandar el evento de detección de movimiento proveniente del sistema de videovigilancia. Se registra el incidente del evento y se vuelve a intentar. - Imposibilidad de conectar con el servidor. Se registra el incidente del evento y se vuelve a intentar. - Imposibilidad de conectar con la aplicación. Se registra el incidente del evento y se vuelve a intentar. - Imposibilidad de crear la notificación para el usuario. Se registra el incidente del evento y se vuelve a intentar. 		

Tabla 18: Caso de uso CU-09

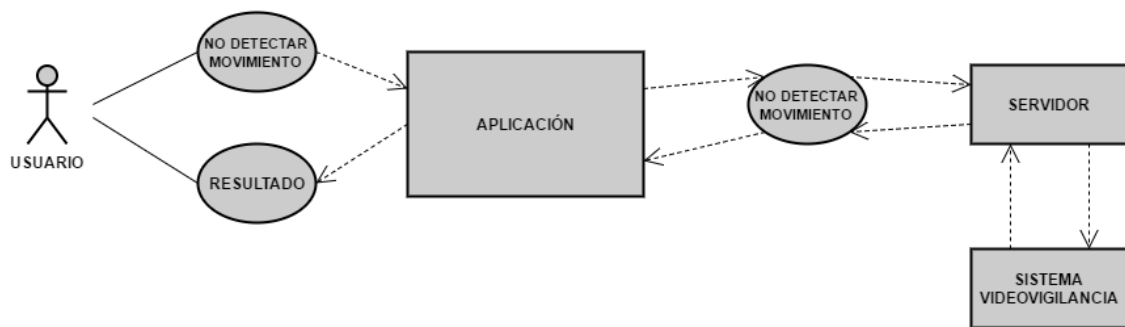


Ilustración 27: Caso de uso CU-10

Identificador	CU-10	Nombre	Desactivar detección movimiento
Actores	Usuario		
Objetivo	El usuario pulsa el botón de desactivar la detección de movimiento		
Escenario	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de desactivar la detección de movimiento. 2. La aplicación recibe la petición y la envía al servidor. 3. El servidor recibe la petición y la manda al sistema de videovigilancia. 4. El sistema de videovigilancia recibe la petición y finaliza la ejecución del software de detección de movimiento. 		
Condiciones de fallo	<ul style="list-style-type: none"> - Imposibilidad de recibir la petición. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de conectar con el servidor. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de conectar con el sistema de videovigilancia. Se muestra un mensaje de error y se recomienda intentar más tarde. - Imposibilidad de finalizar de ejecutar el software de detección de movimiento. Se muestra un mensaje de error y se recomienda intentar más tarde. 		

Tabla 19: Caso de uso CU-10

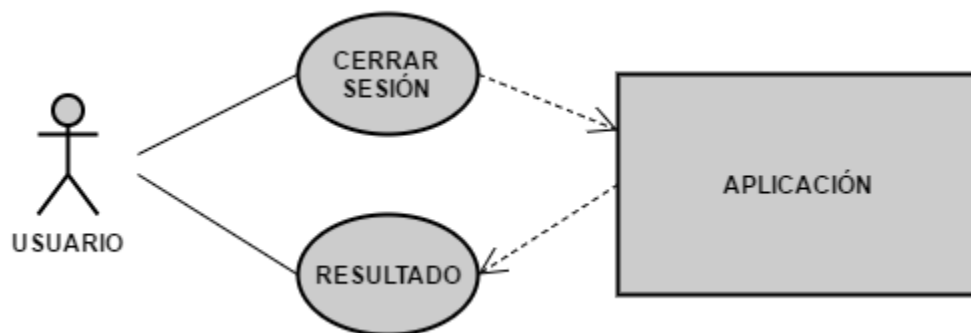


Ilustración 28: Caso de uso CU-11

Identificador	CU-11	Nombre	Salir de la aplicación
Actores	Usuario		
Objetivo	El usuario sale de la aplicación quedando la sesión iniciada.		
Escenario	1. El usuario sale de la aplicación dejando iniciada la sesión		
Condiciones de fallo	Imposibilidad de salir de la aplicación. Se muestra un mensaje de error y se recomienda intentarlo más tarde.		

Tabla 20: Caso de uso CU-11

3.5. Requisitos de Software

Para la definición y descripción de los requisitos de software se va a emplear una serie de tablas, buscando mostrar los requisitos del sistema de manera clara y concisa para su posterior testeo. El formato de la tabla para representar los requisitos es el siguiente:

Identificador		Categoría	
Nombre			
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción			
Requisitos relacionados			

Tabla 21: Formato tabla descripción requisitos software

A continuación se describen los campos que forman la tabla de descripción de requisitos:

- **Identificador**: Este campo representa el código que identifica de manera única al requisito. La nomenclatura que se usará para el nombrado será la siguiente: RX-YY, siendo su significado el siguiente:
 - **R**: Este parámetro indica que se trata de un requisito.
 - **X**: Este parámetro indica a qué grupo pertenece el requisito, puede ser:
 - **F**: Esto indica que el requisito se trata de un Requisito Funcional, describiendo la operación que debe de realizar la aplicación o el sistema.
 - **NF**: Esto indica que el requisito se trata de un Requisito No Funcional, describiendo el cómo se debe de realizar la operación de la aplicación o el sistema.
 - **YY**: Este parámetro indica el número de requisito dentro de la clasificación de requisitos.
- **Categoría**: Este campo representa la categoría correspondiente del requisito.
- **Nombre**: Este campo representa un resumen del requisito formado por un título breve.
- **Prioridad**: Este campo representa el nivel de atención que tiene el requisito y la necesidad que se tiene porque se satisfaga. Dentro de este campo hay 3 divisiones:
 - **Alta**: Este nivel indica que el requisito tiene que ser satisfecho antes que otros requisitos que presenten una prioridad más baja.
 - **Media**: Este nivel indica que el requisito tiene que ser satisfecho después de haber cumplido con los requisitos de nivel Alto.
 - **Baja**: Este nivel indica que el requisito es el último en satisfacerse.
- **Riesgo**: Este campo representa la relación que presenta el requisito con la dificultad que conlleva así como su necesidad dentro del sistema. Dicho campo también presente 3 divisiones de igual manera que el campo anterior.

- **Verificabilidad:** Este campo representa con qué facilidad se puede comprobar qué el requisito se cumple. Dicho campo presenta 3 divisiones:
 - **Alta:** Este nivel indica que el requisito se puede comprobar con todas las herramientas de verificación.
 - **Media:** Este nivel indica que el requisito no se puede comprobar con todas las herramientas de verificación que se puede aplicar.
 - **Baja:** Este nivel indica que el requisito no se puede comprobar con las herramientas que están disponibles.
- **Descripción:** Este campo representa una descripción completa del requisito junto a todos sus aspectos y características.
- **Requisitos relacionados:** Este campo representa los identificadores de los correspondientes requisitos con los que mantiene algún tipo de relación. Este campo en blanco indica que no presenta relación alguna con algún requisito del sistema.

3.3.1. Especificación de Requisitos Funcionales

Identificador	RF-01	Categoría	Sistema
Nombre	Unidad de trabajo		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema trabajará con videos e imágenes provenientes del sistema de videovigilancia además de con los datos de los usuarios.		
Requisitos relacionados	RC-01, RC-03, RC-04		

Tabla 22: Requisito RF-01

Identificador	RF-02	Categoría	Sistema
Nombre	Base de datos		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema utilizará el sistema de gestión de bases de datos relacional MySQL en su versión 5.5.54-0+deb8u1.		
Requisitos relacionados	RC-01, RC-04, RF-01		

Tabla 23: Requisito RF-02

Identificador	RF-03	Categoría	Sistema
Nombre	Campos de los usuarios		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	<p>Los usuarios deberán de tener los siguientes atributos en la base de datos:</p> <ul style="list-style-type: none"> - <u>Nombre de usuario</u>: Este atributo identifica de manera única al usuario de la aplicación del sistema. - <u>Contraseña</u>: Este atributo es la contraseña que da acceso al usuario a la aplicación del sistema. - <u>Código de usuario</u>: Este atributo es un código proporcionado por el sistema para identificar el sistema de videovigilancia. - <u>Dirección de correo electrónico</u>: Este atributo es la dirección de correo electrónico del usuario. - <u>Dirección postal</u>: Este atributo es la dirección postal del usuario. 		
Requisitos relacionados	RC-01, RF-01, RF-02		

Tabla 24: Requisito RF-03

Identificador	RF-04	Categoría	Sistema
Nombre	Inserciones de usuarios		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	Los usuarios deberán de poder ser insertados en la base de datos con todos los campos del formulario de creación de usuarios de la aplicación del sistema.		
Requisitos relacionados	RC-01, RF-01, RF-02, RF-03		

Tabla 25: Requisito RF-04

Identificador	RF-05	Categoría	Sistema
Nombre	Búsqueda de usuarios		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema será capaz de realizar búsquedas sobre todos los campos de los usuarios que dispone.		
Requisitos relacionados	RF-01, RF-02, RF-03, RF-04		

Tabla 26: Requisito RF-05

Identificador	RF-06	Categoría	Sistema
Nombre	Captura de video e imágenes		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema será capaz de capturar vídeo e imágenes del sistema de videovigilancia.		
Requisitos relacionados	RF-01		

Tabla 27: Requisito RF-06

Identificador	RF-07	Categoría	Sistema
Nombre	Retransmisión en vivo		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema permitirá al usuario poder ver la retransmisión en vivo del sistema de videovigilancia.		
Requisitos relacionados	RC-03, RF-01, RF-06		

Tabla 28: Requisito RF-07

Identificador	RF-08	Categoría	Sistema
Nombre	Terminar retransmisión en vivo		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema permitirá al usuario poder terminar la retransmisión en vivo del sistema de videovigilancia.		
Requisitos relacionados	RC-03, RF-01, RF-06, RF-07		

Tabla 29: Requisito RF-08

Identificador	RF-09	Categoría	Sistema
Nombre	Detección de movimiento		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema deberá poder detectar movimiento del sistema de videovigilancia.		
Requisitos relacionados	RF-01, RF-06		

Tabla 30: Requisito RF-09

Identificador	RF-10	Categoría	Sistema
Nombre	Activar detección de movimiento		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema permitirá al usuario activar la detección de movimiento.		
Requisitos relacionados	RC-02, RF-01, RF-06, RF-09		

Tabla 31: Requisito RF-10

Identificador	RF-11	Categoría	Sistema
Nombre	Desactivar detección de movimiento		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema permitirá al usuario desactivar la detección de movimiento.		
Requisitos relacionados	RC-02, RF-01, RF-06, RF-09		

Tabla 32: Requisito RF-11

Identificador	RF-12	Categoría	Sistema
Nombre	Notificación detección de movimiento		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema notificará al usuario si detecta movimiento y enlazará la retransmisión a la notificación.		
Requisitos relacionados	RC-03, RF-01, RF-06, RF-09, RF-10		

Tabla 33: Requisito RF-12

Identificador	RF-13	Categoría	Sistema
Nombre	Generación código aplicación		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema generará un código único de identificación del sistema de videovigilancia y se lo proporcionará al usuario.		
Requisitos relacionados	RC-01, RF-01, RF-02, RF-03		

Tabla 34: Requisito RF-13

Identificador	RF-14	Categoría	Sistema
Nombre	Creación de cuenta de usuario		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema permitirá al usuario crearse una cuenta en la aplicación del sistema de videovigilancia junto al código de aplicación recibido.		
Requisitos relacionados	RC-01, RF-01, RF-02, RF-03, RF-04		

Tabla 35: Requisito RF-14

Identificador	RF-15	Categoría	Sistema
Nombre	Inicio de sesión		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema permitirá al usuario iniciar sesión en la aplicación del sistema de videovigilancia con su nombre de usuario y su contraseña.		
Requisitos relacionados	RC-01, RF-01, RF-02, RF-03, RF-05		

Tabla 36: Requisito RF-15

Identificador	RF-16	Categoría	Sistema
Nombre	Cierre de sesión		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema permitirá al usuario cerrar sesión en la aplicación del sistema de videovigilancia.		
Requisitos relacionados	RC-04, RF-15		

Tabla 37: Requisito RF-16

Identificador	RF-17	Categoría	Sistema
Nombre	Salir de la aplicación		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema permitirá al usuario salir de la aplicación del sistema de videovigilancia.		
Requisitos relacionados	RC-04, RF-15		

Tabla 38: Requisito RF-17

Identificador	RF-18	Categoría	Sistema
Nombre	Detección de movimiento segundo plano		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema deberá continuar la detección de movimiento en segundo plano si el usuario sale de la aplicación.		
Requisitos relacionados	RF-1, RF-06, RF-09, RF-10, RF-15, RF-17		

Tabla 39: Requisito RF-18

Identificador	RF-19	Categoría	Sistema
Nombre	Registro de movimiento		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	<p>El sistema deberá registrar el evento de la detección de movimiento con el siguiente formato: DD/MM/YY HH:MM:SS, siendo los campos:</p> <ul style="list-style-type: none"> - <u>DD</u>: Este campo representa el día del evento. - <u>MM</u>: Este campo representa el mes del evento. - <u>YY</u>: Este campo representa el año del evento. - <u>HH</u>: Este campo representa la hora del evento. - <u>MM</u>: Este campo representa los minutos del evento. - <u>SS</u>: Este campo representa los segundos del evento. 		
Requisitos relacionados	RF-1, RF-09, RF-10		

Tabla 40: Requisito RF-19

Identificador	RF-20	Categoría	Sistema
Nombre	Almacenamiento de vídeo o imágenes		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema deberá almacenar vídeo o imágenes cuando se produzca el evento de detección de movimiento en el sistema de videovigilancia.		
Requisitos relacionados	RF-1, RF-06, RF-09, RF-10, RF-19		

Tabla 41: Requisito RF-20

3.3.2. Especificación de Requisitos No Funcionales

Identificador	RNF-01	Categoría	Interfaz
Nombre	Framework de desarrollo		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La interfaz del sistema deberá de realizarse utilizando el framework de desarrollo que ofrece Android Studio.		
Requisitos relacionados	RR-02, RR-03		

Tabla 42: Requisito RNF-01

Identificador	RNF-02	Categoría	Interfaz
Nombre	Idioma principal interfaz		
Prioridad	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La interfaz del sistema tendrá como idioma principal el español castellano.		
Requisitos relacionados	RR-01		

Tabla 43: Requisito RNF-02

Identificador	RNF-03	Categoría	Interfaz
Nombre	Idioma secundario interfaz		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	La interfaz del sistema tendrá como idioma secundario el inglés.		
Requisitos relacionados	RR-01, RNF-02		

Tabla 44: Requisito RNF-03

Identificador	RNF-04	Categoría	Operatividad
Nombre	Usuario administrador		
Prioridad	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema necesitará un usuario administrador que sea el encargado del control, del mantenimiento y de las incidencias del sistema de videovigilancia.		
Requisitos relacionados	RF-01		

Tabla 45: Requisito RNF-04

Identificador	RNF-05	Categoría	Interfaz
Nombre	Elementos de la interfaz, inicio de sesión		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	<p>El sistema dispondrá de una interfaz presentándose los siguientes apartados dentro de ella:</p> <ul style="list-style-type: none"> - Iniciar sesión - Crear cuenta - Salir 		
Requisitos relacionados	RR-02, RF-14, RF-15, RF-17		

Tabla 46: Requisito RNF-05

Identificador	RNF-06	Categoría	Interfaz
Nombre	Elementos de la interfaz, pantalla de inicio		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	<p>El sistema dispondrá de una interfaz presentándose los siguientes apartados dentro de ella:</p> <ul style="list-style-type: none"> - Ver en vivo - Detectar movimiento - Cerrar sesión - Salir 		
Requisitos relacionados	RR-02, RF-07, RF-09, RF-10, RF-11, RF-12, RF-17, RF-18		

Tabla 47: Requisito RNF-06

Identificador	RNF-07	Categoría	Interfaz
Nombre	Elementos de la interfaz, retransmisión en vivo		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema dispondrá de una interfaz presentándose los siguientes apartados dentro de ella: <ul style="list-style-type: none"> - Video en vivo - Terminar video en vivo 		
Requisitos relacionados	RR-02, RF-08		

Tabla 48: Requisito RNF-07

Identificador	RNF-08	Categoría	Interfaz
Nombre	Elementos de la interfaz, creación de cuenta		
Prioridad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Riesgo	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Descripción	El sistema dispondrá de una interfaz presentándose los siguientes campos a rellenar dentro de ella: <ul style="list-style-type: none"> - Nombre de usuario - Contraseña - Código de usuario - Dirección de correo electrónico - Dirección postal 		
Requisitos relacionados	RR-02, RF-14		

Tabla 49: Requisito RNF-08

3.4. Matriz de Trazabilidad

En este apartado se va a proceder a realizar un análisis para encontrar alguna posible inconsistencia, ambigüedad, duplicidad o escasez de información. Este análisis se realiza sobre los requisitos de usuario y sobre los requisitos de software, de forma que se asocian entre ellos para determinar funcionalidades y comportamientos comunes.

Para llevarlo a cabo se va a proceder a construir una matriz de trazabilidad entre los requisitos de usuario y los requisitos de software (que son los requisitos funcionales y los requisitos no funcionales).

Con la construcción de esta matriz, se comprueba que todos los requisitos del sistema que se han especificado son válidos, consistentes y completos, así como que existe por lo menos un requisito de usuario por cada requisito de software definido anteriormente.

A continuación se muestra la construcción de la segunda matriz de trazabilidad:

	RC-01	RC-02	RC-03	RC-04	RR-01	RR-02	RR-03
RF-01	X		X	X			
RF-02	X			X			
RF-03	X						
RF-04	X						
RF-05	X						
RF-06			X				
RF-07			X				
RF-08			X				
RF-09			X				
RF-10		X					
RF-11		X					
RF-12			X				
RF-13	X						
RF-14	X						
RF-15	X						
RF-16				X			
RF-17				X			
RF-18			X				
RF-19			X				
RF-20			X				
RNF-01						X	X
RNF-02					X		
RNF-03					X		
RNF-04	X						
RNF-05					X		
RNF-06					X		
RNF-07					X		
RNF-08					X		

Tabla 50: Matriz trazabilidad Requisitos Usuario/Requisitos Software

A simple vista se puede observar que los requisitos de usuario y los requisitos de software completan la funcionalidad que se viene buscando. Esto se debe a que todos requisitos de usuario tienen al menos un requisito de software asociado, de la misma manera los requisitos de software tienen al menos un requisito de usuario.

4. Diseño del Sistema

A lo largo de este apartado se va a llevar a cabo un diseño que se encargará de definir la arquitectura que presenta el sistema, así como se definirá de igual manera la tecnología que le dará forma al sistema y se procederá a especificar con más detalle los distintos componentes de los que está formado el sistema.

En primer lugar, se va a proceder a explicar de forma detallada cada uno de los componentes de los que forman el sistema, describiendo sus funciones, el diseño de las clases, el diseño físico de los datos, las interfaces de usuario, la forma que presentarán los ficheros y las bases de datos.

4.1. Diseño de la Arquitectura

En este apartado se describirán los distintos módulos que formarán los subsistemas de soporte utilizados por el sistema desarrollado. El diseño del sistema se corresponderá con el patrón de diseño Cliente-Servidor, ya que se define una organización independiente por parte del Cliente (dispositivos Android) y por parte del Servidor [10]:

- **Cliente**: Este subsistema será el encargado en participar en el establecimiento de las conexiones, enviando una petición al Servidor esperando por la respuesta de este. Presenta un tiempo de vida finito ya que una vez que su solicitud es procesada, su trabajo termina.
- **Servidor**: Este subsistema será el encargado de ofrecer servicios a través de la red. Acepta las peticiones, realiza el servicio correspondiente y devuelve el resultado obtenido al Cliente que envió la petición. Puede ejecutarse en cualquier sistema en el que exista TCP/IP junto a otros programas de aplicación. La ejecución del Servidor comienza antes de las interacciones con el Cliente siendo su tiempo de vida infinito.

Este patrón de arquitectura permite proveer al usuario de cierto tipo de libertad para obtener la información deseada en un momento dado proveniente de fuentes locales o distantes, procesándola como convenga. Los servidores también pueden intercambiar información dentro de esta arquitectura.

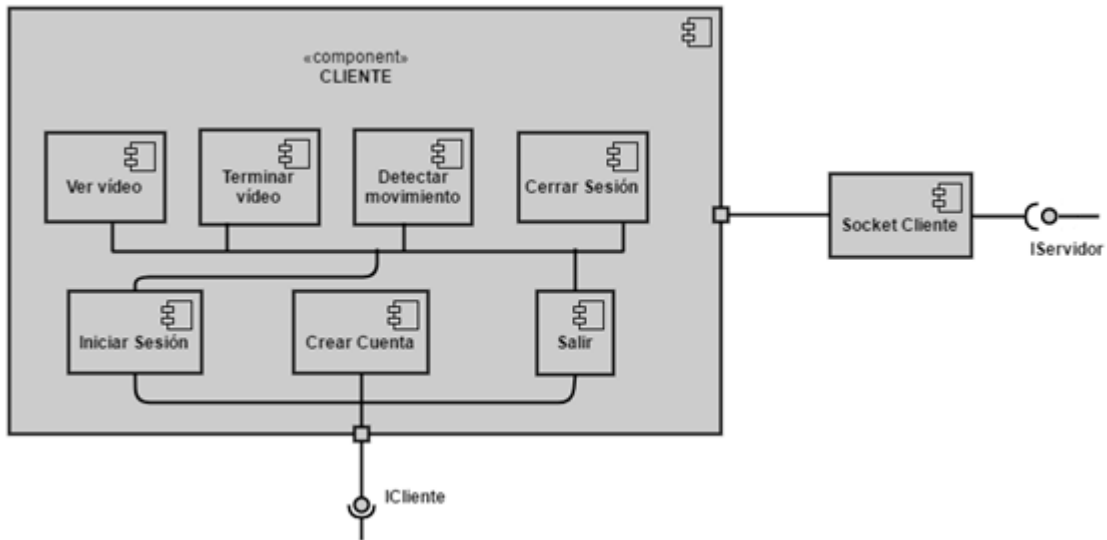


Ilustración 29: Modelo Cliente-Servidor [11]

4.1.1. Cliente

Como se ha comentado en el apartado anterior, este subsistema es el encargado de establecer las conexiones con el Servidor, a través de diferentes tipos de peticiones. Contiene el núcleo de las peticiones por parte del usuario, así como las herramientas necesarias para enviar peticiones y procesar las respuestas recibidas por parte del Servidor.

En el sistema es el dispositivo con la plataforma Android, jugará el papel de Cliente dentro del sistema.



En esta ilustración se pueden apreciar los componentes que forman el subsistema del Cliente:

- **Iniciar Sesión**: Este componente del Cliente es el encargado de permitir o no permitir la sesión del usuario dentro de la aplicación del sistema de videovigilancia, con los datos del nombre de usuario y contraseña del usuario de la aplicación.
- **Crear Cuenta**: Este componente del Cliente es el encargado de permitir al usuario de la aplicación crearse una nueva cuenta junto al código de aplicación proporcionado por el sistema para el correcto registro. Necesita la siguiente información por parte del usuario:
 - Nombre de usuario
 - Contraseña de usuario
 - Confirmación de contraseña de usuario
 - Código de aplicación
 - Dirección de correo electrónico
 - Dirección postal
- **Salir**: Este componente del Cliente es el encargado de permitir al usuario salir de la aplicación.

- **Ver Vídeo:** Este componente del Cliente es el encargado de permitir al usuario con sesión iniciada ver en vivo el vídeo proveniente del sistema de videovigilancia.
- **Terminar Vídeo:** Este componente del Cliente es el encargado de permitir al usuario con sesión iniciada y con la retransmisión del vídeo en vivo iniciada de terminar la retransmisión del video en vivo proveniente del sistema de videovigilancia.
- **Detectar movimiento:** Este componente del Cliente es el encargado de permitir al usuario con sesión iniciada tanto activar como desactivar la detección de movimiento por parte del sistema de videovigilancia.
- **Cerrar Sesión:** Este componente del Cliente es el encargado de permitir al usuario con sesión iniciada cerrar la sesión de usuario en la aplicación.
- **Socket Cliente:** Este componente del Cliente es el encargado de comunicarse con el Servidor, tanto para enviar la petición del usuario desde la aplicación del dispositivo con plataforma Android, como de recibir y de procesar la respuesta proveniente del Servidor. El Socket Cliente a veces es llamado desde la aplicación sin embargo cuando trabaja en segundo plano no es necesario utilizar la interfaz de la aplicación.

Todos los componente definidos del subsistema del Cliente, tienen que pasar por el Socket Cliente para enviar la petición deseada al Servidor, procesar la respuesta y crear la interfaz de Cliente correspondiente a la respuesta recibida.

4.1.2. Servidor

Como se ha comentado en el apartado anterior, este subsistema es el encargado de recibir las peticiones provenientes del Cliente, procesarlas y enviar una respuesta. Contiene el núcleo del funcionamiento del sistema, ya que es el encargado de retransmitir el vídeo al cliente y de detectar movimiento en el sistema, así como de crear notificaciones y registrar los eventos de movimiento que se produzcan en el sistema de videovigilancia.

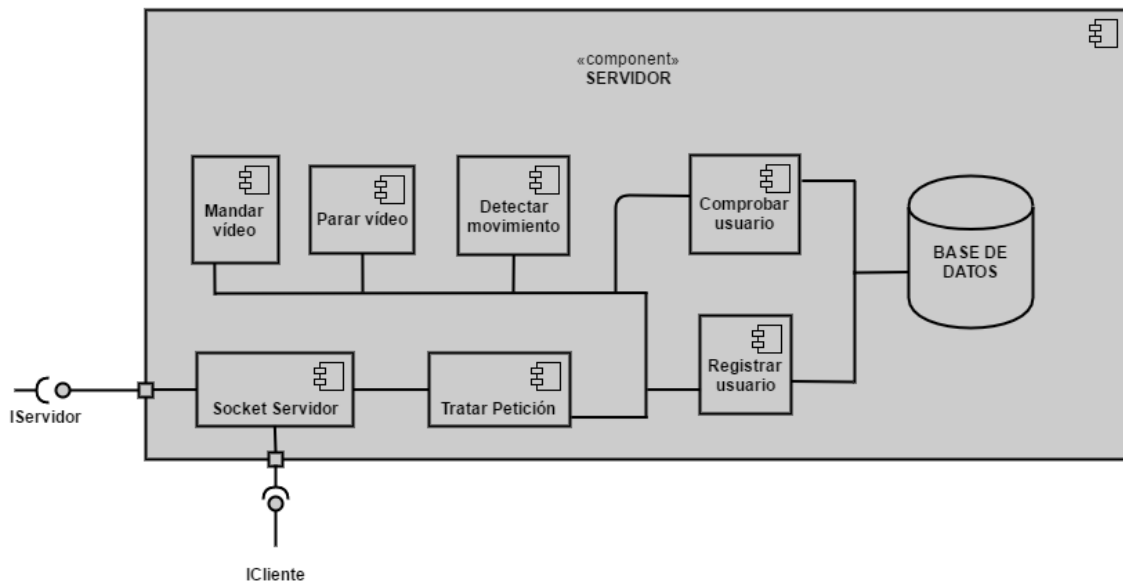


Ilustración 31: Servidor

En esta ilustración se pueden apreciar los componentes que forman el subsistema del Servidor:

- **Socket Servidor**: Este componente del Servidor es el encargado de recibir las peticiones por parte del Cliente y mandar dicha petición al componente Tratar Petición.
- **Tratar Petición**: Este componente del Servidor es el encargado de recibir la petición del Socket Servidor y posteriormente realizar su tratamiento, llamando a los componentes que sean necesarios.
- **Comprobar Usuario**: Este componente del Servidor es el encargado de recibir el nombre de usuario y la contraseña de un usuario y comprobar dichos datos dentro de la base de datos del sistema.
- **Registrar Usuario**: Este componente del Servidor es el encargado de recibir los datos provenientes del formulario de registro de un usuario e insertarlos en la base de datos del sistema.

- **Mandar Vídeo:** Este componente del Servidor es el encargado de mandar el vídeo proveniente del sistema de videovigilancia para su posterior retransmisión en vivo.
- **Parar Vídeo:** Este componente del Servidor es el encargado de parar de mandar el vídeo proveniente del sistema de videovigilancia para dejar de retransmitir en vivo.
- **Detectar Movimiento:** Este componente del Servidor es el encargado de detectar movimiento en el sistema de videovigilancia además de avisar, registrar y capturar el evento producido.

4.2. Diseño de Clases

En este apartado se va a proceder a crear el modelo de clases de diseño del sistema. Dicho modelo estará formado por todos los detalles y especificaciones de cada una de las clases, teniendo en cuenta los escenarios que se han contemplado en los casos de uso y en la arquitectura que se ha definido en los apartados anteriores.

4.2.1. Diagrama de Clases

La arquitectura en la que se basa el sistema sigue el patrón Cliente-Servidor, se procede a diferenciar las clases que forman parte tanto del subsistema Cliente como del subsistema Servidor, reflejando la separación en el diagrama de clases del sistema.

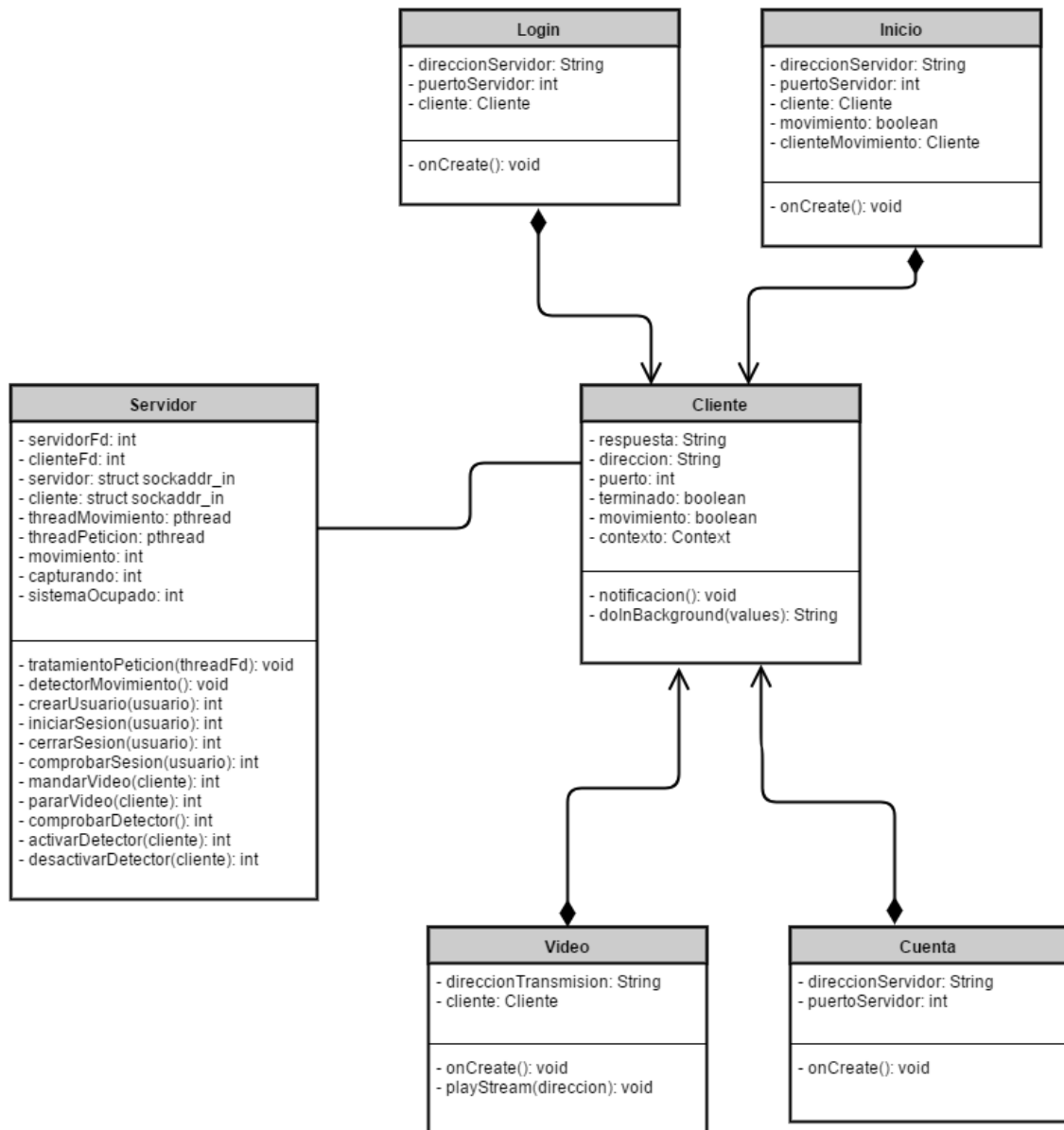


Ilustración 32: Diagrama de clases

Se puede apreciar a simple vista qué clases forman el subsistema del Servidor y qué clases forman el subsistema del Cliente. Cabe destacar que en la parte del Servidor todo el tratamiento de peticiones y sus respectivos métodos para tratarlas se concentran en la misma clase. Sin embargo en la parte del Cliente no es así, ya que cada clase corresponde con una parte de la aplicación y a su vez cada clase implementa la interfaz de la parte en sí.

Todas las variables y los métodos relacionados con la implementación de la interfaz se han omitido en este diagrama a fin de simplificar y mostrar de forma más precisa la implementación del sistema.

4.2.2. Identificación de Atributos y Métodos

El diagrama de clases anteriormente definido representa las clases junto a sus atributos y métodos del sistema. Sin embargo en este apartado se va a proceder a detallar en profundidad cada una de las clases que definen el sistema, sin añadir como se comentó anteriormente aquellas clases, métodos y atributos que forman parte de la interfaz del sistema.

Para ello se va a seguir un formato tabular similar al formato elegido para la definición tanto de los casos de uso como de los requisitos del sistema. Dicho formato es el siguiente:

CL-XX	
Nombre de la clase	
Descripción	
Atributos	
Métodos	

Tabla 51: Formato tabla definición Clases

Los campos de este formato tabular que definen las clases del sistema son los siguientes:

- **CL-XX**: Este campo representa el código único que identifica a cada una de las clases. La nomenclatura que se procede a usar es la siguiente: CL-XX, donde XX es el número que representa a la clase.
- **Nombre de la clase**: Este campo representa el nombre de la clase que se procede a definir.
- **Descripción**: Este campo representa un breve resumen de la finalidad de la clase.
- **Atributos**: Este campo representa la lista con los atributos que contiene la clase.
- **Métodos**: Este campo representa la lista con los métodos que contiene la clase.

A continuación se muestran las definiciones de las clases del sistema:

CL-01	
Nombre de la clase	Servidor
Descripción	Esta clase encierra el Servidor que se encarga de recibir las peticiones del Cliente, de procesar dichas peticiones y de elaborar una respuesta para enviársela.
Atributos	<ul style="list-style-type: none"> - int servidorFd - int clienteFd - struct sockaddr_in servidor - struct sockaddr_in cliente - pthread threadMovimiento - pthread threadPetición - int movimiento - int capturando - int sistemaOcupado
Métodos	<ul style="list-style-type: none"> - void tratamientoPetición(threadFd) - void detectorMovimiento() - int crearUsuario(usuario) - int iniciarSesion(usuario) - int cerrarSesion(usuario) - int comprobarSesion(usuario) - int mandarVideo(cliente) - int pararVideo(cliente) - int comprobarDetector() - int activarDetector(cliente) - int desactivarDetector(cliente)

Tabla 52: Clase CL-01

CL-02	
Nombre de la clase	Cliente
Descripción	Esta clase encierra el Cliente que se encarga de mandar las peticiones del usuario de la aplicación al Servidor, así como de recoger las respuestas de este.
Atributos	<ul style="list-style-type: none"> - String respuesta - String direccion - int puerto - boolean terminado - boolean movimiento - Context contexto
Métodos	<ul style="list-style-type: none"> - void notificacion() - String doInBackground(values)

Tabla 53: Clase CL-02

CL-03	
Nombre de la clase	Login
Descripción	Esta clase se encarga del inicio de sesión en la aplicación por parte del usuario, recogiendo el usuario y la contraseña introducido para comunicarse con la clase Cliente y así crear y mandar la petición al Servidor, esperando su respuesta.
Atributos	<ul style="list-style-type: none"> - String direccionServidor - int puertoServidor - Cliente
Métodos	void onCreate()

Tabla 54: Clase CL-03

CL-04	
Nombre de la clase	Inicio
Descripción	<p>Esta clase se encarga de mostrar al usuario las funcionalidades que puede realizar:</p> <ul style="list-style-type: none"> - Ver vídeo - Detectar movimiento - Cerrar sesión - Salir de la aplicación <p>Cuando el usuario seleccione alguna de estas opciones, se comunicará con la clase Cliente y así crear una petición y mandarla al Servidor, esperando su respuesta. Además comprobará en todo momento si se ha detectado movimiento en el sistema.</p>
Atributos	<ul style="list-style-type: none"> - String direccionServidor - int puertoServidor - Cliente - boolean movimiento - Cliente clienteMovimiento
Métodos	void onCreate()

Tabla 55: Clase CL-04

CL-05	
Nombre de la clase	Video
Descripción	<p>Esta clase se encarga de mostrar al usuario la retransmisión en vivo por parte del sistema de videovigilancia, comunicándose con la clase Cliente para crear y mandar la petición al Servidor, esperando la respuesta.</p>
Atributos	<ul style="list-style-type: none"> - String direccionTransmision - Cliente
Métodos	<ul style="list-style-type: none"> - void onCreate() - void playStream(String dirección)

Tabla 56: Clase CL-05

CL-06	
Nombre de la clase	Cuenta
Descripción	<p>Esta clase se encarga de permitir al usuario crearse una cuenta en la aplicación del sistema mostrándole unos campos que tiene que rellenar:</p> <ul style="list-style-type: none"> - Nombre de usuario - Contraseña de usuario - Confirmación de contraseña de usuario - Código de aplicación - Dirección de correo electrónico - Dirección postal
Atributos	<ul style="list-style-type: none"> - String direccionServidor - int puertoServidor
Métodos	void onCreate()

Tabla 57: Clase CL-06

4.2.3. Diagramas de Secuencia

En este apartado se van a proceder a realizar una serie de diagramas de secuencia para visualizar las interacciones que se dan con el sistema en cada uno de los casos de uso que se han definido en el apartado 3.2.2. Especificación de los Casos de Uso.

- Diagrama de Secuencia del caso de uso CU-01. Comprobar estado del sistema.

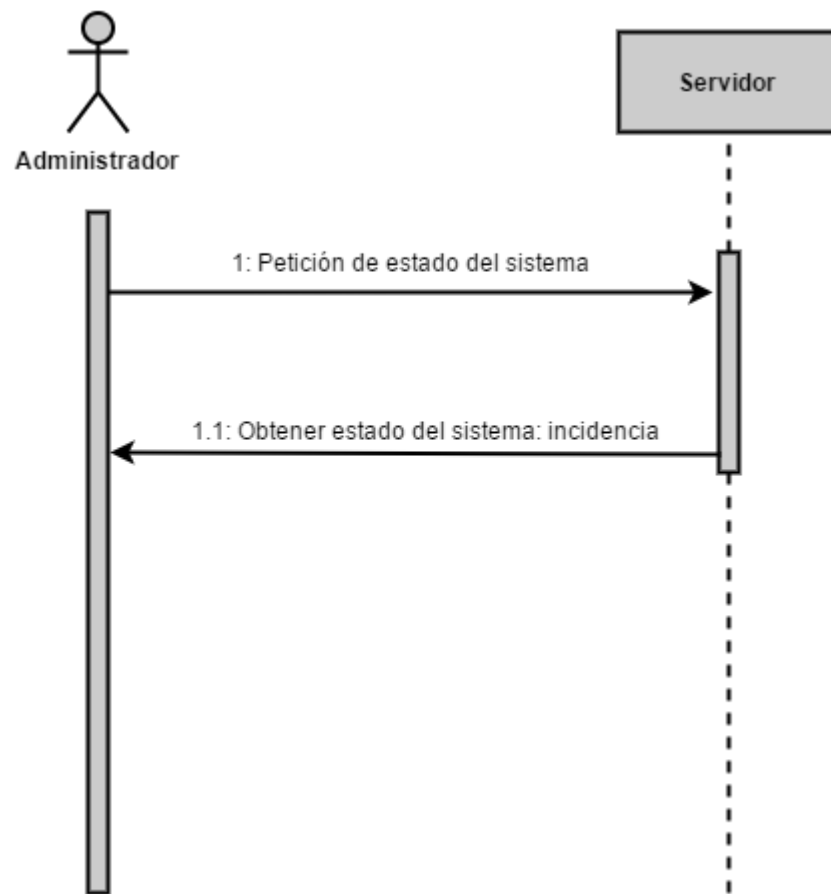


Ilustración 33: Diagrama Secuencia CU-01

- Diagrama de Secuencia del caso de uso CU-02. Resolver incidencia del sistema.

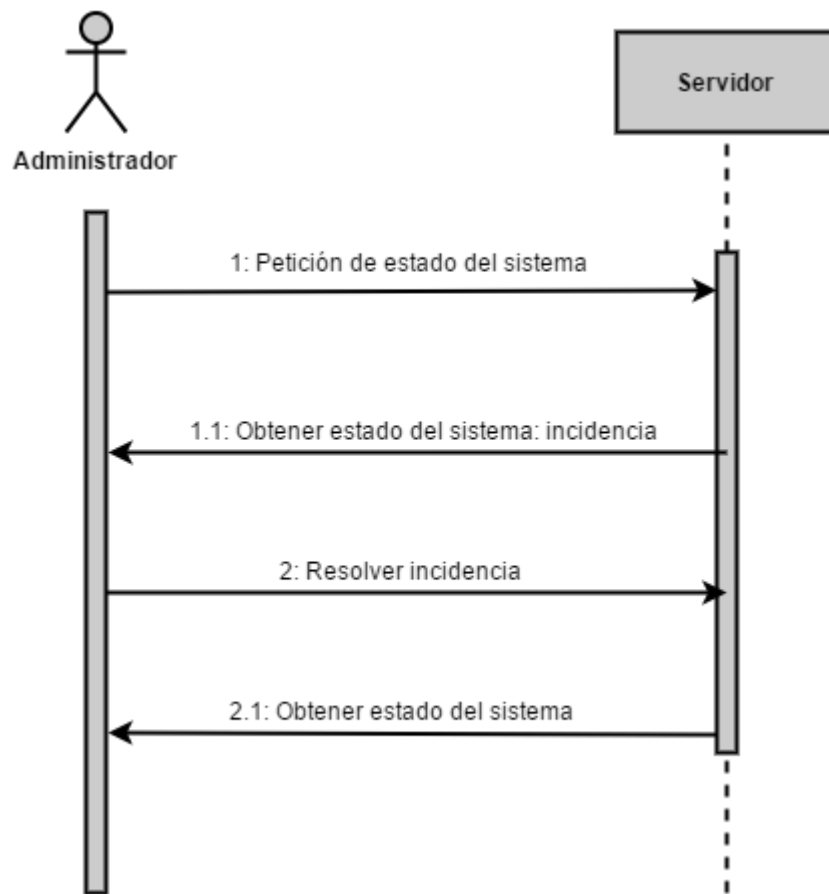


Ilustración 34: Diagrama Secuencia CU-02

- Diagrama de Secuencia del caso de uso CU-03. Crear una cuenta de usuario.

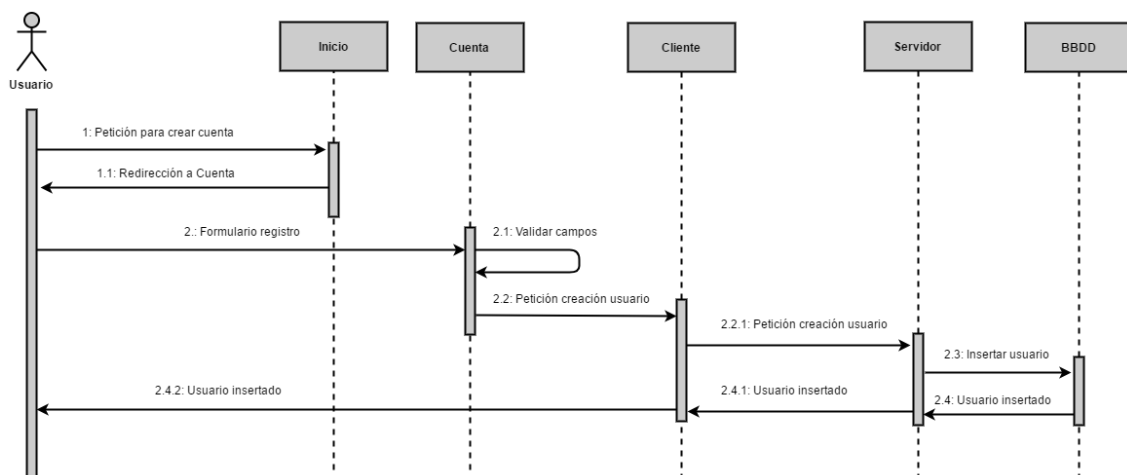


Ilustración 35: Diagrama Secuencia CU-03

- Diagrama Secuencia del caso de uso CU-04. Iniciar sesión en la aplicación.

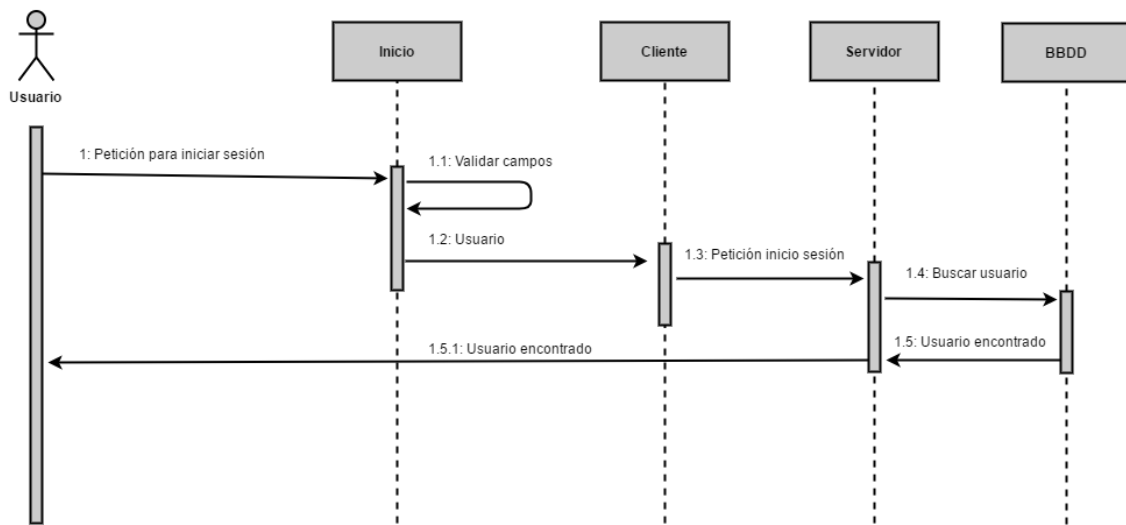


Ilustración 36: Diagrama Secuencia CU-04

- Diagrama Secuencia del caso de uso CU-05. Cerrar sesión en la aplicación.

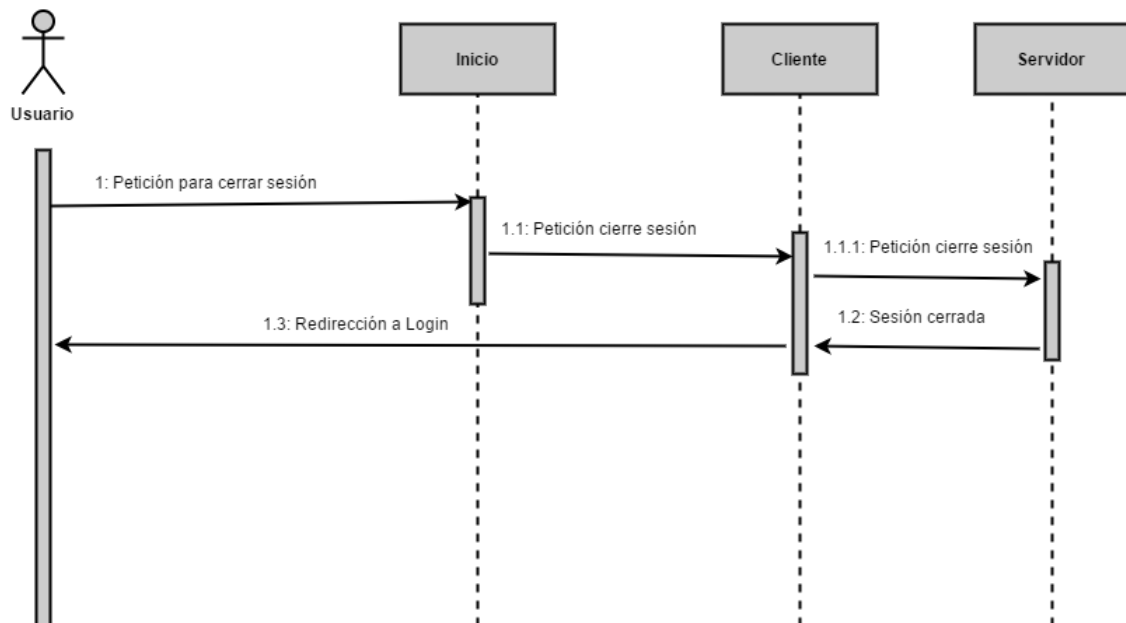


Ilustración 37: Diagrama Secuencia CU-05

- Diagrama Secuencia para el caso de uso CU-06. Ver vídeo.

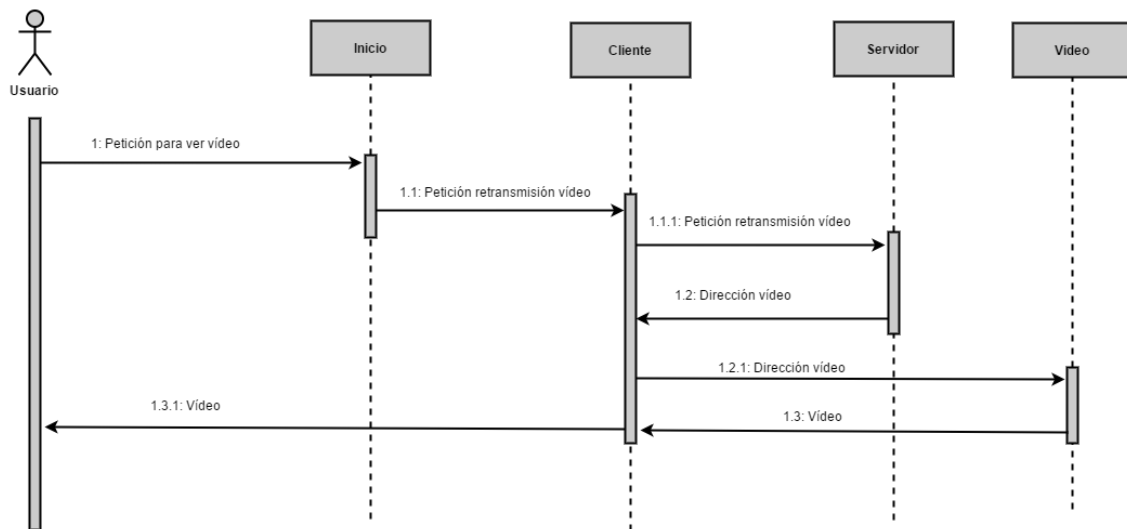


Ilustración 38: Diagrama Secuencia CU-06

- Diagrama Secuencia para el caso de uso CU-07.

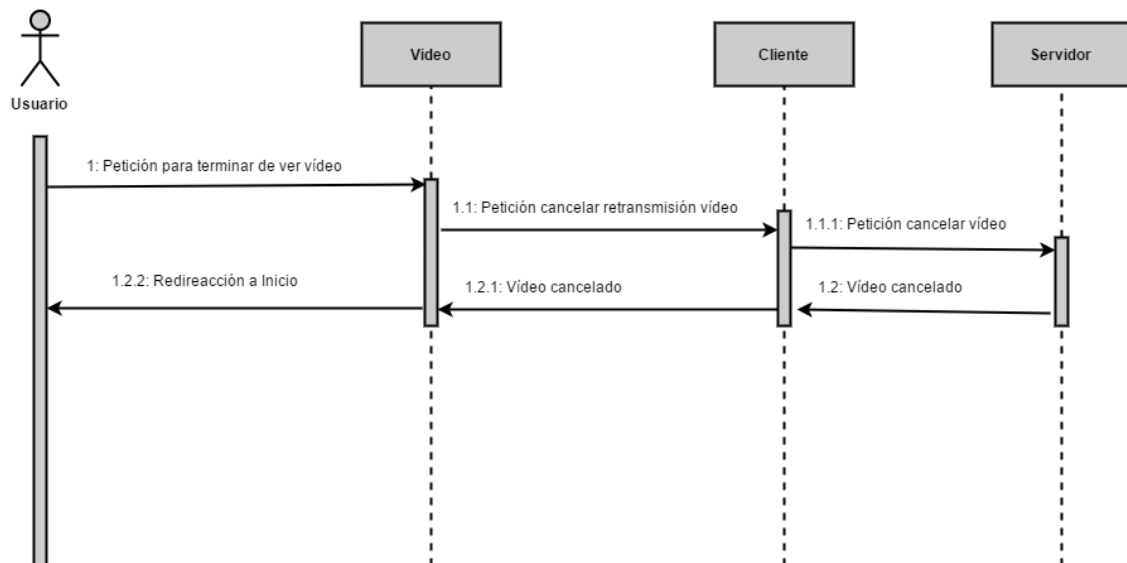


Ilustración 39: Diagrama Secuencia CU-07

- Diagrama Secuencia para el caso de uso CU-08

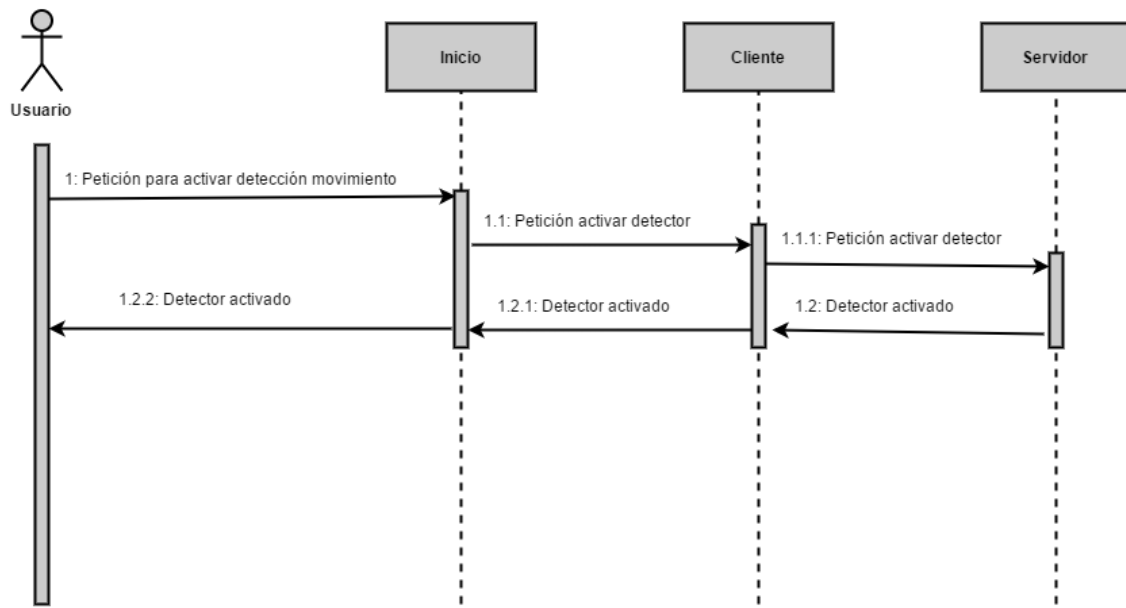


Ilustración 40: Diagrama Secuencia CU-08

- Diagrama Secuencia para el caso de uso CU-09

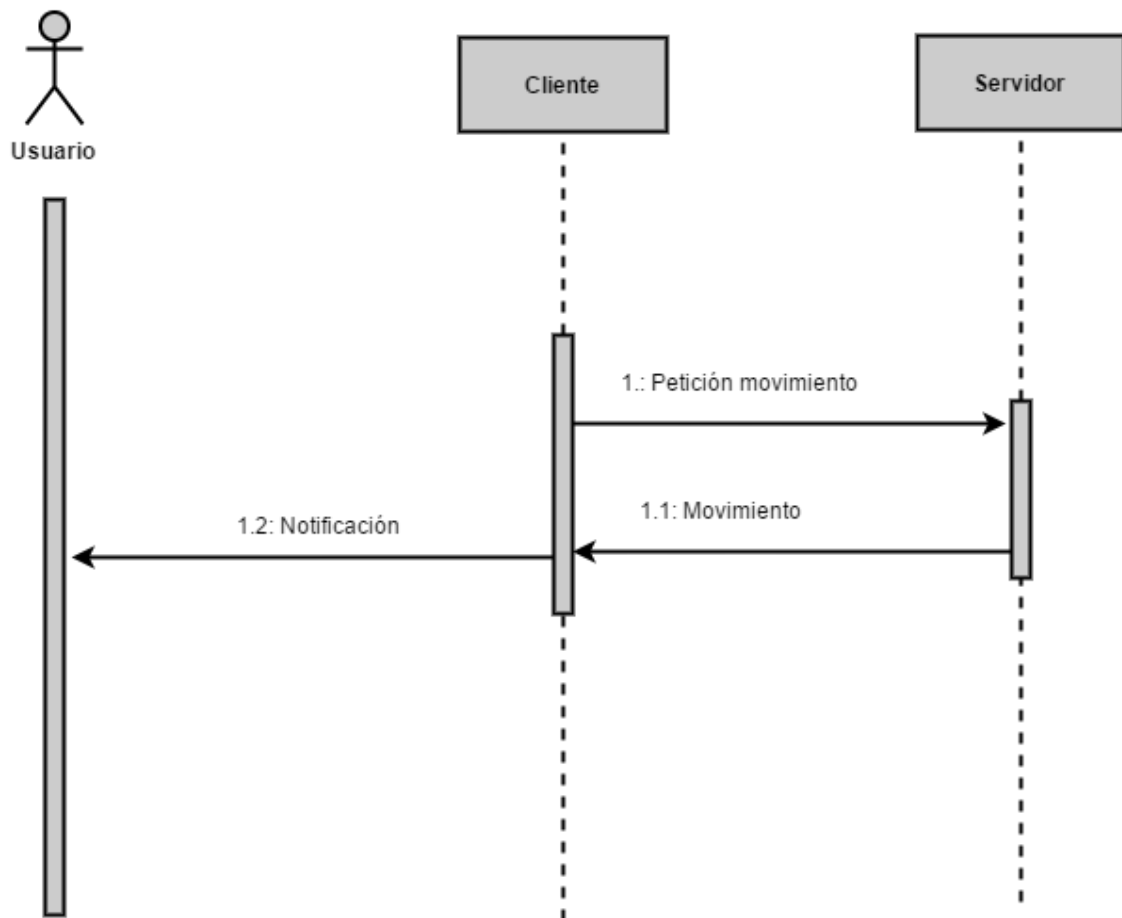


Ilustración 41: Diagrama Secuencia CU-09

- Diagrama Secuencia para el caso de uso CU-10

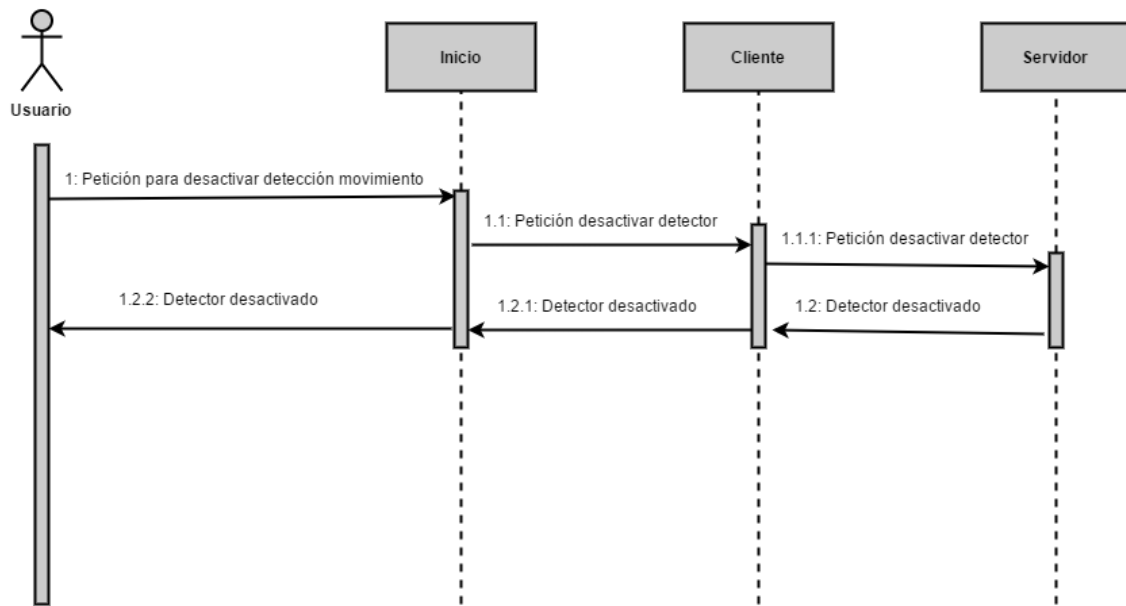


Ilustración 42: Diagrama Secuencia CU-10

- Diagrama Secuencia para el caso de uso CU-11

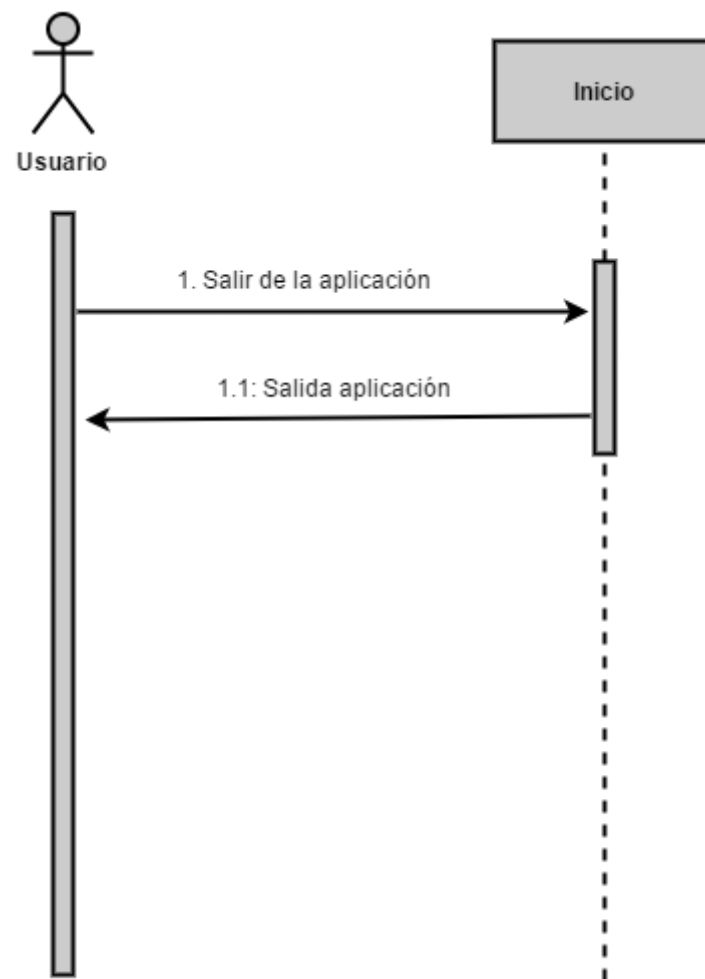


Ilustración 43: Diagrama Secuencia CU-11

4.3. Diseño Físico de Datos

En este apartado se va a proceder a especificar el diseño físico de datos, en el que se definen las estructuras físicas de los datos que se van a utilizar en el sistema.

Para llevar a cabo dicho diseño, se van a tener en cuenta los requisitos de software que se han definido en la fase de análisis del sistema, de igual manera que las características del entorno tecnológico para tratar los datos de la manera correcta respecto del sistema.

4.3.1. Base de Datos

Como se ha comentado en la parte de los requisitos del sistema, la base de datos del sistema se creará y se gestionará con el gestor de bases de datos MySQL. Dicha base de datos se almacenará en el subsistema del Servidor donde este podrá acceder a ella y modificar su contenido y el Cliente a través de las peticiones podrá realizar tareas de consulta y de inserción de usuarios.

Dicha base de datos presenta las siguientes características:

Atributo	Tipo	Especialidad
nombreUsuario	Varchar(32)	Primary Key
contrasena	Varchar(32)	-
codigoApp	Varchar(64)	-
direccionCorreo	Varchar(32)	Primary Key
direccionPostal	Varchar(32)	-

Cabe destacar que las respectivas contraseñas de los usuarios del sistema no se guardan en claro dentro de la base de datos, se utiliza un cifrado SHA-256 para aportar robusted y seguridad tanto a la aplicación como a los datos personales. De igual manera la dirección del correo electrónico y la dirección postal del usuario también se cifrarán bajo SHA-256 puesto que son datos personales del usuario.

El nombre de usuario será la clave primaria de la base de datos junto a la dirección de correo electrónico, puesto que suponemos el posible caso de instalar varios sistemas de videovigilancia pero de distinto usuario. El código de la aplicación es generada por el sistema y es única para identificar a los distintos sistemas que hay, por lo que en la generación del código ya se contempla esa acción y no es necesario establecer dicho atributo como clave primaria.

Por lo tanto la dupla nombre de usuario y dirección de correo electrónico serán la clave primaria de la base de datos del sistema.

4.4. Diseño de Interfaz de Usuario

En este apartado se procederá a definir las interfaces de usuarios con las que interactuará al usar la aplicación del sistema una vez que el subsistema Servidor esté en funcionamiento.

El sistema presentará las siguientes interfaces de usuario:

4.4.1. Página de Login

Esta interfaz del sistema visualizará las opciones para el usuario de iniciar sesión en la aplicación, crear una cuenta de la aplicación o salir de la aplicación.

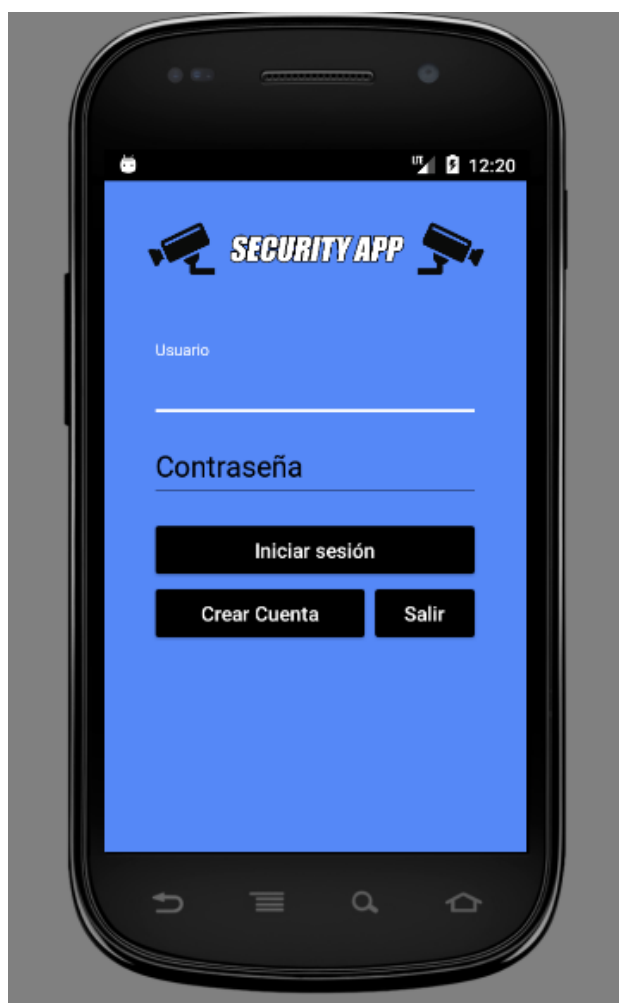


Ilustración 44: Interfaz Login Sistema Vertical

A simple vista podemos ver las funcionalidades que nos aporta esta interfaz del sistema, las cuales se han definido en los requisitos: iniciar sesión en la aplicación, crearse una cuenta de la aplicación, salir de la aplicación. Nos muestra dos campos a rellenar en caso de que queramos iniciar sesión en la aplicación.

El estilo de esta interfaz será el que se utilice para las demás interfaces, con el contenido de la sección en primer lugar, estando las diferentes funcionalidades que se presenten en la sección en la zona inferior a la del contenido principal de la sección. De esta forma se busca la sencillez y la precisión de cara a la cercanía con el usuario, ya que si rotamos el dispositivo la interfaz se ajusta añadiendo scroll cuando sea necesario.

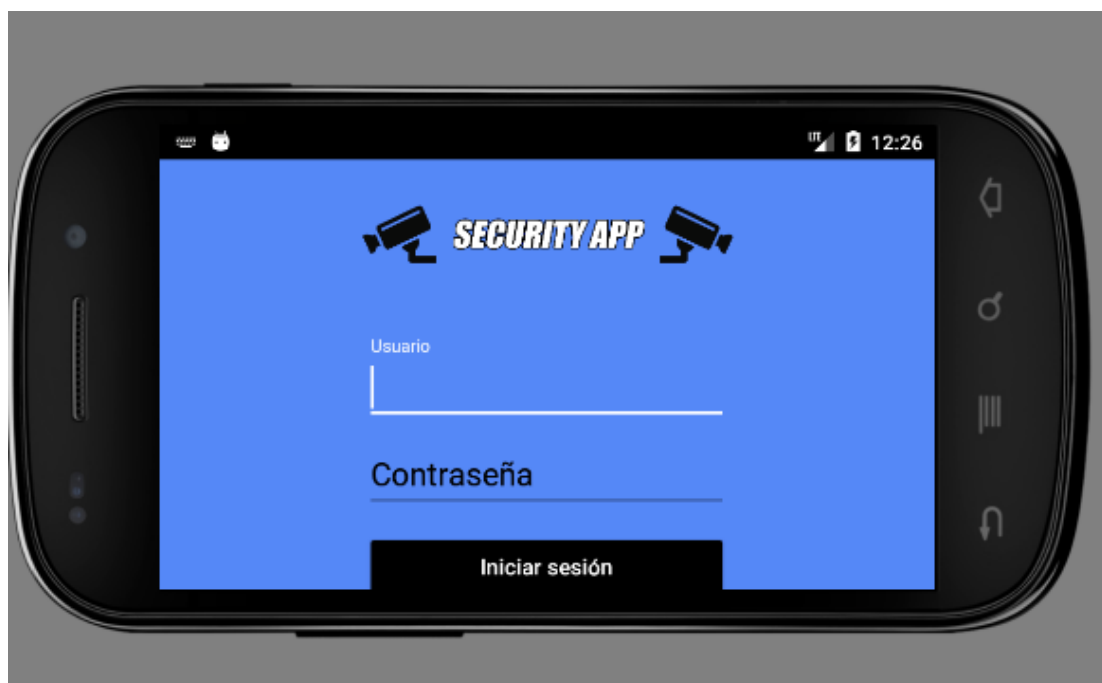


Ilustración 45: Interfaz Login Sistema Horizontal 1

De esta manera a simple vista no podemos observar todo el contenido de la interfaz, sin embargo se da la posibilidad al usuario de realizar scroll para visualizar todo el contenido.

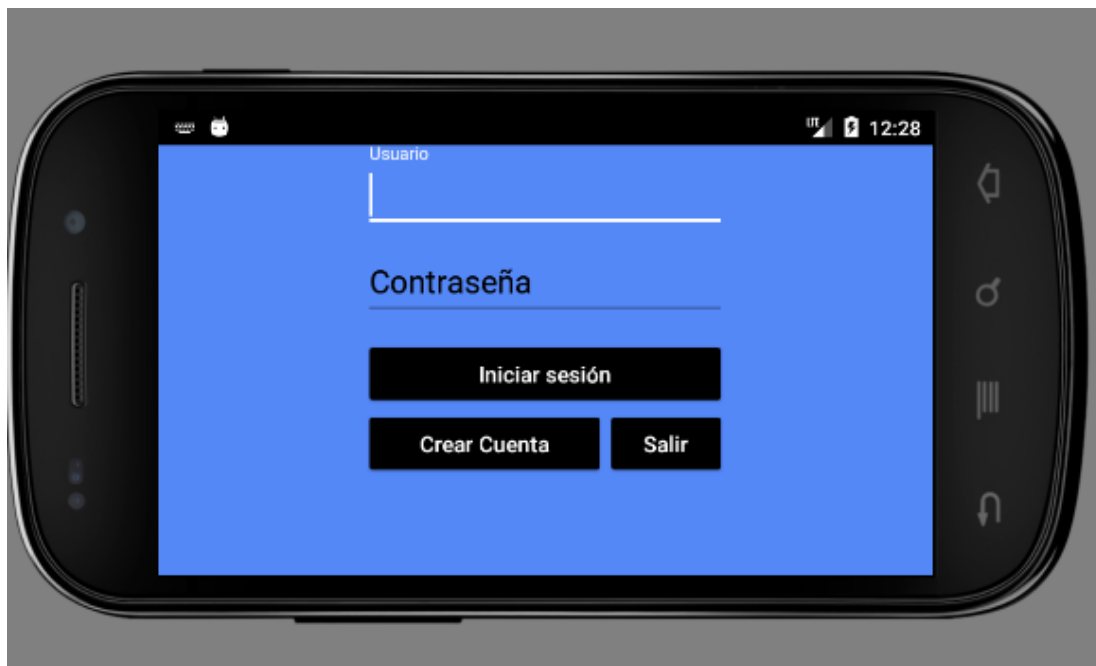


Ilustración 46: Interfaz Login Sistema Horizontal 2

4.4.2. Página de Crear Cuenta

Esta sección es accedida por el usuario cuando pretende crearse una cuenta en la aplicación. Una vez obtenga el código de la aplicación proporcionado por el sistema, puede crearse una cuenta en la aplicación:



Ilustración 47: Interfaz Crear Cuenta Sistema Vertical 1



Ilustración 48: Interfaz Crear Cuenta Sistema Vertical 2

Dicha sección está formada por todos aquellos campos que deben de ser rellenos por el usuario con el formato correcto para crearse la cuenta en la aplicación. Dispone de la función de volver a la sección anterior de igual manera. Con el dispositivo en horizontal la interfaz ofrece las mismas funcionalidades.



Ilustración 49: Interfaz Crear Cuenta Sistema Horizontal 1



Ilustración 50: Interfaz Crear Cuenta Sistema Horizontal 2

4.4.2. Página de Inicio

Esta sección es accedida por el usuario una vez que ha iniciado sesión en la aplicación con una cuenta creada anteriormente:



Ilustración 51: Interfaz Inicio Sistema Vertical

Podemos observar que se le permite al usuario detectar movimiento en el sistema para que sea avisado en caso de detectar movimiento, se le permite ver en vivo las imágenes del sistema de videovigilancia, de igual modo se le permite al usuario tanto cerrar sesión en la aplicación como salir de la aplicación. Todas las funcionalidades contempladas en

los requisitos del sistema especificados. De igual manera en horizontal utilizando en este caso scroll para visualizar todo el contenido.



Ilustración 52: Interfaz Inicio Sistema Horizontal 1

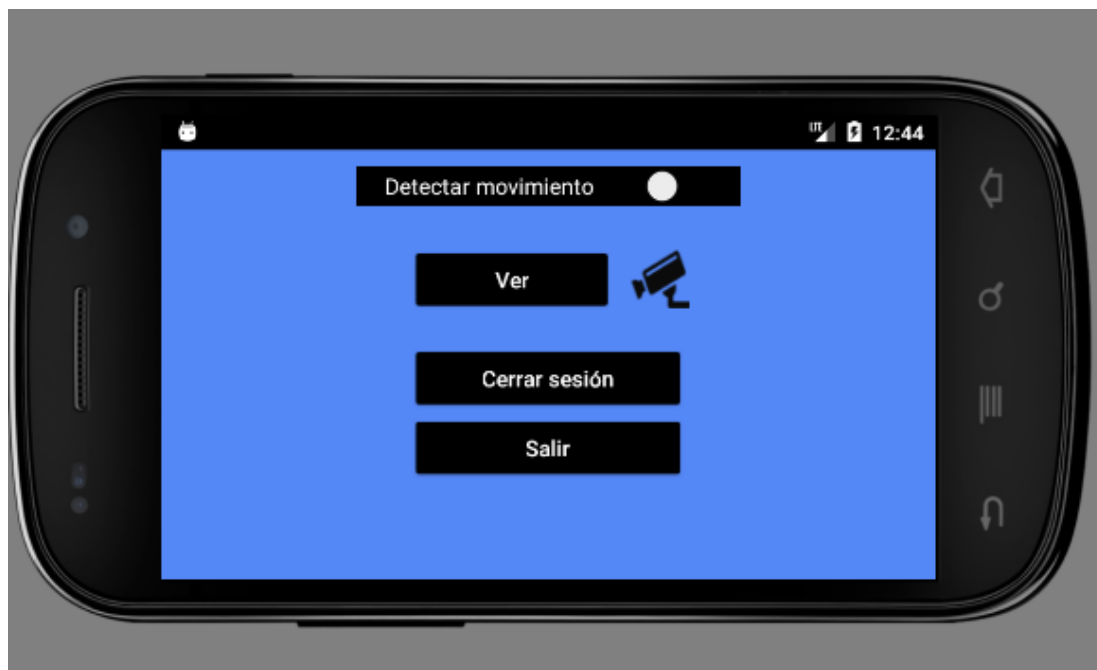


Ilustración 53: Interfaz Inicio Sistema Horizontal 2

4.4.3. Página de Vídeo

Esta sección es accedida con un usuario con sesión iniciada en la aplicación, cuando el usuario desea ver el contenido del vídeo e imágenes del sistema de videovigilancia:



Ilustración 54: Interfaz Video Sistema Vertical

Se puede observar el contenido del sistema de videovigilancia así como la función de terminar de ver el vídeo. La interfaz en horizontal intenta aprovechar el espacio para mostrar en mayor tamaño el contenido del vídeo, incluyendo un scroll.

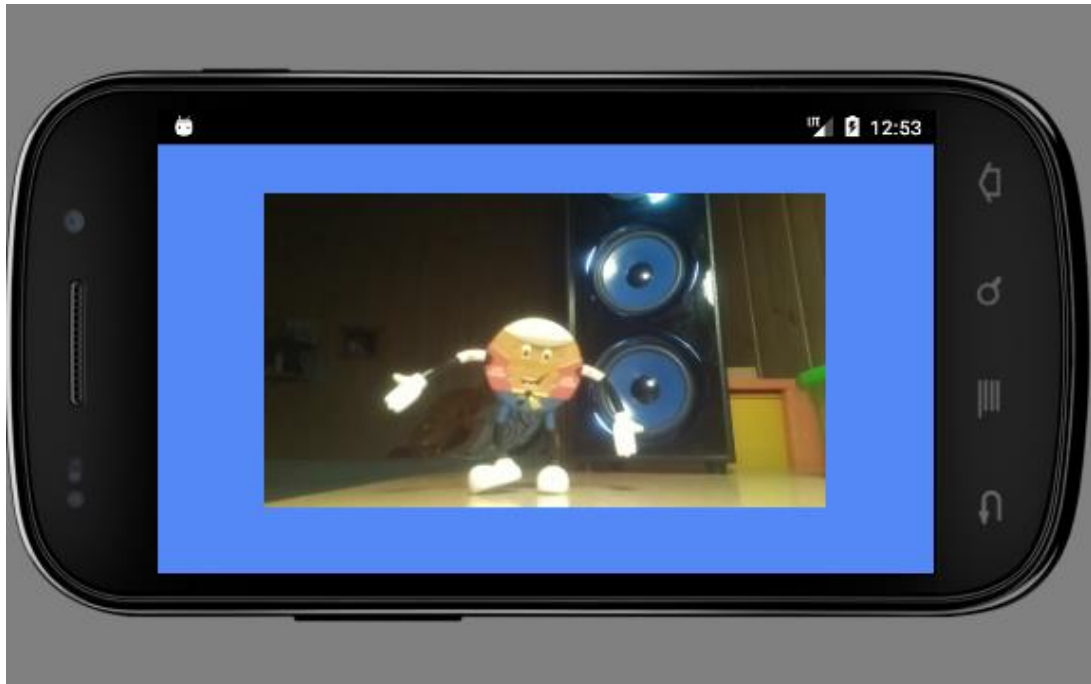


Ilustración 55: Interfaz Video Sistema Horizontal 1

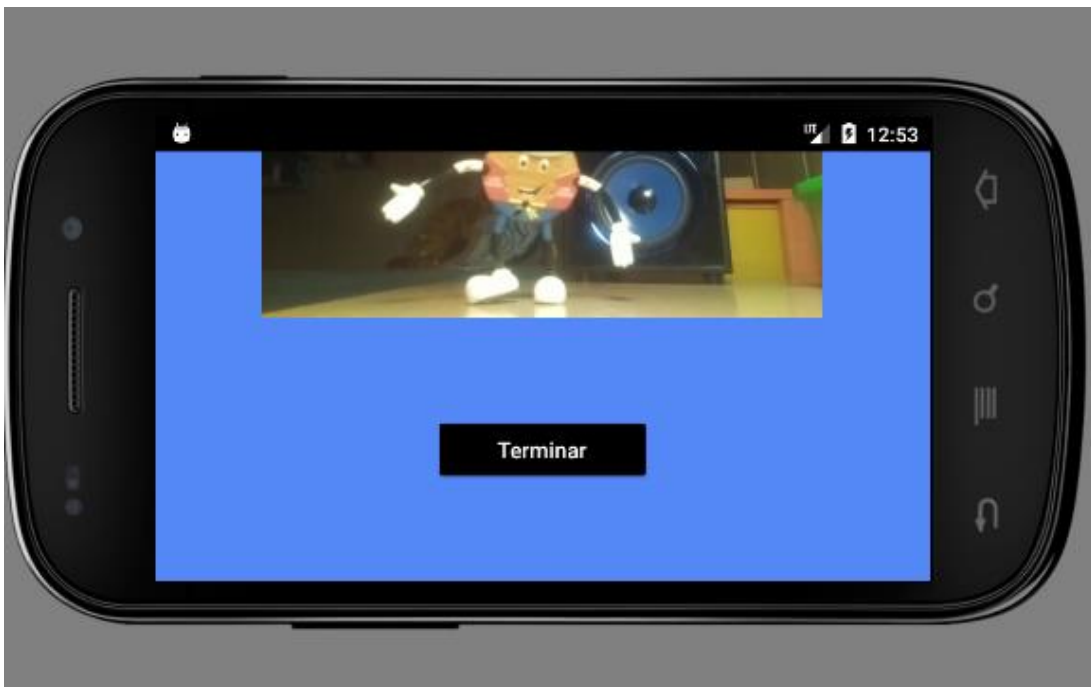


Ilustración 56: Interfaz Video Sistema Horizontal 2

4.5. Especificación del entorno de Desarrollo

En este apartado se procederá a clasificar las diversas herramientas que se han utilizado durante el proceso de desarrollo del sistema. Se ha debido de tomar decisiones antes de la elección final de dichas herramientas así como el lenguaje de programación a usar, software para la comunicación de los subsistemas. Para ello esta sección se dividirá en diferentes secciones.

4.5.1. Hardware

Los dispositivos informáticos que se han utilizado para el desarrollo del proyecto tanto para su creación como para la realización de pruebas, son los siguientes:

Modelo	Personal Computer
Procesador	2.7 GHz Intel Core i5 6400
Memoria RAM	8 GB 2133 MHz DDR4 G.Skill Ripjaws V Red
Disco Duro	120 GB SSD Toshiba Q300
Tipo Sistema	Sistema operativo de 64 bits, procesador x64
Sistema Operativo	Windows 10 Pro Versión 1703

Tabla 58: Configuración ordenador 1

Modelo	Raspberry Pi 2 Model B
Procesador	900MHz quad-core ARM Cortex-A7 CPU
Memoria RAM	1 GB
Disco Duro	8 GB SD
Tipo Sistema	Sistema operativo de 64 bits, procesador x64
Sistema Operativo	Raspbian GNU/Linux Jessie Versión 8

Tabla 59: Configuración ordenador 2

Modelo	BQ Aquaris M5
Procesador	1.5 GHz Qualcomm Snapdragon 615 Octa Core A53
Memoria RAM	2 GB
Disco Duro	16 GB ROM
Tipo Sistema	Sistema operativo móvil de 64 bits, procesador x64
Sistema Operativo	Android Marshmallow Versión 6.0.1

Tabla 60: Configuración dispositivo móvil

4.5.2. Sistemas Operativos

Para el desarrollo del proyecto se han utilizado diferentes sistemas operativos los cuales serán expuestos en este apartado, especificando las funcionalidades por las que se han escogido cada uno de ellos. Dichas decisiones se basan en la disponibilidad, en el conocimiento de uso, en la compatibilidad con el entorno de desarrollo tanto en el proyecto como en la realización de pruebas y funcionamiento en los diferentes dispositivos.

4.5.2.1 Windows 10

Windows 10 es un sistema operativo desarrollado por Microsoft siendo uno de los más conocidos en el mercado además de utilizados. Puede ser instalado en cualquier dispositivo si este es compatible con licencia de pago. La razón de uso de dicho sistema operativo es además de porque es que estaba instalado en el equipo, por su conocimiento de uso y por su compatibilidad a la hora de utilizar las herramientas para desarrollar el sistema.

4.5.2.2 Raspbian Jessie 8

Raspbian Jessie es un sistema operativo compatible con las Raspberry Pi 2 Model B más utilizado. Está basado en GNU/Linux y consta de una interfaz sencilla así como de multitud de herramientas. La razón de uso de dicho sistema operativo es porque está basado en GNU/Linux lo cual es conocido además de ser un sistema operativo ligero para el dispositivo Raspberry Pi 2 Model B, permitiendo acceder a todos los módulos instalados del dispositivo de manera precisa y sencilla.

4.5.2.3 Marshmallow 6.0.1

Marshmallow es una de las versiones del sistema operativo utilizado en los dispositivos Android compatibles. El uso de un sistema operativo móvil es debido a la realización de pruebas tanto de funcionamiento como de interfaz gráfica del sistema.

4.5.3 Lenguajes de programación

En este apartado se procederá a enunciar los diferentes lenguajes de programación que se han utilizado para el desarrollo del proyecto y de las herramientas que conllevan su correcto funcionamiento.

4.5.3.1 Java

Java es un lenguaje de programación así como una plataforma informática que se comercializó por primera vez en el 1995 a manos de Sun Microsystems [23]. Muchas aplicaciones y sitios web no funcionarían de no ser porque Java está instalado. Java presenta las características de ser rápido, de ser seguro y de ser fiable. Su uso está orientado a centros de datos, consolas, juegos, súper computadoras, teléfonos móviles e incluso el mismo Internet.

Java ya ha sido utilizado durante los años anteriores en los diferentes proyectos que se han ido realizando. Se dispone de experiencia así como de conocer el entorno de desarrollo escogido: Android Studio.

4.5.3.2 C#

C# es también conocido como “Lenguaje de programación de sistemas”. Este lenguaje de programación fue desarrollado en el año 1972 a manos de Dennis Ritchie para el sistema operativo multiplataforma UNIX [24]. Es un tipo de lenguaje de alto nivel sin embargo con este lenguaje de programación se permite la posibilidad de programar a bajo nivel. Este lenguaje de programación presenta librerías que incluyen funciones para el desarrollo de los programas, siendo un lenguaje de programación compacto y sencillo.

C# ya ha sido utilizado durante los años anteriores en los diferentes proyectos que se han ido realizando, aplicándolo en arquitecturas Cliente-Servidor de ahí su elección para implementar la parte del subsistema Servidor.

4.5.3.3 Python

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows hasta servidores de red. El código no es necesario compilarlo para ejecutarlo, por lo que es un lenguaje de programación rápido a la hora de desarrollar. Su creador fue Guido Van Rossum, con la intención de cubrir la necesidad de un lenguaje orientado a objetos sencillo para tratar tareas diversas dentro de la programación de entonces. Dicho lenguaje de programación es multiplataforma, interpretado, interactivo, orientado a objetos, con funciones y librerías presentando una sintaxis clara [25].

Este lenguaje de programación apenas se ha utilizado durante los años anteriores, sin embargo Raspbian Jessie presenta unas herramientas basadas en Python y en OpenCV para el acceso al módulo de la cámara así como de tratamiento de imágenes, por las que se ha decantado el uso de este lenguaje de programación, orientado a la detección de movimiento del sistema de videovigilancia.

4.5.4 Lenguajes de marcas

El desarrollo de una aplicación orientada al usuario conlleva que se usarán diferentes tipos de herramientas entre las que se encuentran lenguajes que no se consideran de programación, llamados lenguajes de marcas que se han utilizado para el desarrollo del sistema más enfocado a la interfaz.

4.5.4.1 XML

XML viene del inglés eXtensive Markup Language, es un lenguaje de etiquetas manejando cada paquete de información delimitándose por dos etiquetas, separando el contenido de la presentación. Se plantea como un lenguaje estándar para el intercambio

de información entre diferentes programas de forma segura, fiable y libre ya que dicho lenguaje no pertenece a ninguna compañía [26].

Su campo de aplicación se orienta a las bases de datos, los documentos de texto, las hojas de cálculo y las páginas web. Dicho lenguaje es extensible, analizador estándar, facilidad a la hora de realizar análisis y procesar documentos XML de terceros. Además permite relacionar aplicaciones de diferentes lenguajes y plataformas explotando así su universalidad [27].

XML ya se ha utilizado durante los años anteriores en los diferentes proyectos que se han ido realizando, siendo la herramienta que utiliza Android Studio para realizar las interfaces de las aplicaciones.

4.5.5. Software de Desarrollo

En este apartado se va a proceder a incluir todo el software que se haya utilizado para el desarrollo del proyecto así como las aplicaciones de escritorio cuyos servicios se han utilizado para el desarrollo del proyecto.

4.5.5.1 Android Studio

Android Studio es un IDE basado en IntelliJ IDEA de la compañía JetBrains, proporcionando mejoras con respecto al plugin ADT para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma.

Fue presentado por Google el 16 de mayo de 2013, con el objetivo de crear un entorno dedicado en exclusiva para la programación de aplicaciones para dispositivos Android. El 8 de diciembre de 2014 sale la versión estable de Android Studio 1.0, proporcionando soporte para programar aplicaciones Android Wear, herramientas Lint, usa ProGuard además de Gradle, posibilita el control de versiones entre otras muchas funcionalidades que presenta este entorno de desarrollo [28].

En este caso se va a proceder a utilizar Android Studio en su versión 2.3.1 para el desarrollo e implementación de la aplicación Android que realizará la función de Cliente en la arquitectura del sistema.

4.5.5.2 Geany

Geany es un editor de texto ligero que está basado en Scintilla con características básicas de IDE. Se encuentra disponible para distintos sistemas operativos como GNU/Linux, Mac OS X o Windows. Es distribuido como software libre bajo la GPL de GNU.

Presenta soporte para muchos lenguajes de programación distintos como C#, Java, Python entre otros. Presenta las características de autocompletado, soporte multidocumento, coloreado de sintaxis, buscador integrado y herramientas para compilar permitiendo ejecutar directamente desde el entorno además de la descomposición y representación de las clases y estructuras del código [29].

En este caso se va a utilizar Geany en su versión 1.24.1 para el desarrollo de la parte de Python orientada a la detección de movimiento como el desarrollo de la parte del Servidor usando C#.

4.5.5.3 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Dicho sistema de gestión nos permite a través de una serie de sentencias tener información almacenada en una base de datos y recuperarla en el momento que se necesite de una manera rápida y eficiente, permitiendo la creación de bases de datos así como de tablas con atributos donde se almacena la información [30].

En este caso se va a utilizar MySQL en su versión 5.5.54-0+deb8u1, será una herramienta utilizada para el tratamiento de los datos de los usuarios de la aplicación así como de los registros como de los inicios de sesión dentro de la aplicación.

4.5.5.4 VLC

VLC Media Player es un reproductor multimedia, gratuito y multiplataforma. Soporta un gran número de formatos de audio sin necesidad de instalar códecs a MPEG-1, MPEG-2, MPEG-4, DivX, MP3, OGG, MOV, RAM, AVI, FLV, entre otros. Se puede utilizar como servidor de streaming en una red local o de banda ancha.

En este caso se va a utilizar para retransmitir el vídeo procedente del sistema de videovigilancia en una dirección determinada para que el cliente pueda visualizarlo a través de la aplicación Android [31].

4.6. Diseño del Circuito Electrónico del Sistema de Videovigilancia

En este apartado se va a proceder a mostrar el circuito electrónico del sistema de videovigilancia así como sus componentes, el cual se muestra a continuación:

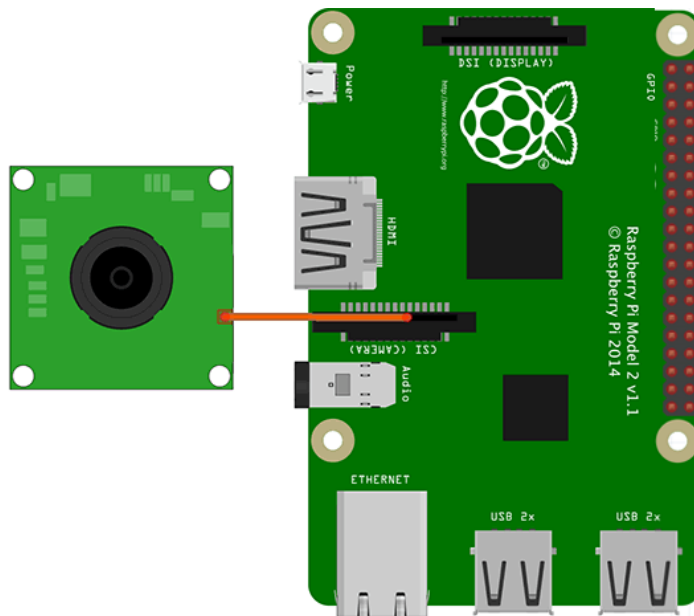


Ilustración 57: Circuito Electrónico Sistema Videovigilancia

Dicho circuito como se puede observar está formado por la Raspberry Pi 2 Model B junto al módulo asociado que es la cámara que procederá a capturar las imágenes o vídeo del entorno donde se haya instalado.

Como se puede observar es un sistema sencillo, en el que se incluiría la fuente de alimentación para dar corriente a la Raspberry Pi 2 Model B, siendo el consumo de este muy pequeño en comparación con otros dispositivos como puede ser un ordenador.

5. Implementación y Plan de Pruebas

En este apartado se va a llevar a cabo la descripción de la implementación de la arquitectura y del diseño del sistema, aspectos detallados en los apartados anteriores. Se va a proceder a mostrar los aspectos de mayor importancia en la fase de desarrollo después del análisis y el diseño del sistema, mostrando el tipo de tecnologías usadas para implementar el proyecto.

Se continuará con la descripción del proceso que se va a seguir para comprobar que se están cumpliendo con todas las funcionalidades especificadas que se han definido en los apartados 3 y 4, análisis y diseño del sistema respectivamente, teniéndose en cuenta los requisitos, casos de uso y la arquitectura.

Para ello se procederá a definir el alcance y el entorno de las pruebas para posteriormente pasar a especificar dichas pruebas del sistema conociendo qué se va a probar y sobre qué entorno.

5.1. Tecnologías Utilizadas

De tal manera como se mencionó en los apartados anteriores, el subsistema Cliente que va a ser la aplicación en la plataforma Android será realizada con el lenguaje de programación Java usando el entorno de desarrollo Android Studio, mientras que el subsistema Servidor será realizada en su mayor parte con el lenguaje de programación C# y Python, utilizando Geany como entorno de desarrollo.

El subsistema Cliente puede ser ejecutado en cualquier dispositivo Android compatible mientras que el subsistema Servidor puede ser ejecutado en cualquier sistema operativo compatible y con las librerías GNU/Linux.

Finalmente la base de datos del sistema se realizará con el gesto de bases de datos MySQL la cual se alojará junto al subsistema Servidor.

5.1.1. Cliente

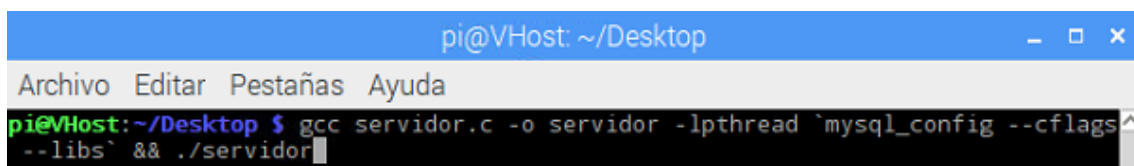
La parte del subsistema Cliente se ha desarrollado con Android Studio, el cual utiliza el lenguaje de programación Java para desarrollar las aplicaciones, usando XML para el desarrollo e implementación de las interfaces de la aplicación. También ofrece la posibilidad de crear dispositivos Android virtuales donde ir probando la aplicación y mostrar sus resultados y funcionalidades actuales, cuya herramienta es muy útil tanto para testear como para observar las funcionalidades presentes en el desarrollo del momento de la aplicación. Para ejecutar esta parte simplemente instalamos el archivo .apk generado de la aplicación en el dispositivo de pruebas, se instala y se usa la aplicación.

5.1.2. Servidor

Geany es un editor de texto con el que se ha llevado a cabo el desarrollo y la implementación del subsistema Servidor. Esta parte en su mayoría se ha realizado con el lenguaje de programación C# para crear la parte Servidor que reciba las peticiones del Cliente, tratarlas y enviar una respuesta. Dicho servidor es un servidor multihilo creando un hilo por cada petición recibida, el cual la trata y elabora la respuesta al Cliente.

También se ha usado Geany para implementar la parte de detección de movimiento, usando en este caso el lenguaje de programación Python, el cual permite acceder de forma sencilla y precisa a los componentes de la Raspberry Pi 2 Model B y el tratamiento de imágenes y vídeos usando la librería OpenCV, de forma rápida y sencilla.

Para ejecutar el Servidor del sistema, se procede a ejecutar el siguiente comando:



```
pi@VHost: ~/Desktop
Archivo  Editar  Pestañas  Ayuda
pi@VHost:~/Desktop $ gcc servidor.c -o servidor -lpthread `mysql_config` --cflags --libs` && ./servidor
```

Se compila el archivo del servidor con los flags necesarios para incorporar las herramientas para poder usar los hilos y la base de datos del sistema. Una vez ejecutado el comando el servidor estará listo para recibir peticiones del cliente.

5.1.3. Hardware

Como se ha mencionado y mostrado en apartados anteriores, la parte hardware del sistema es muy simple, la cual está formada por la Raspberry Pi 2 Model B, el módulo de cámara, fuente de alimentación y un conector HDMI para poder manejar la Raspberry Pi 2 Model B. En este último aspecto se ha utilizado la herramienta TeamViewer para manejarlo todo desde el ordenador sin necesidad del conector HDMI.

Simplemente la Raspberry está conectada a la corriente con un cable de carga USB, conectada a internet con un cable Ethernet, y el módulo de la cámara conectada a la Raspberry. En este sistema se puede incluir un botón que haga la función de apagado/encendido del dispositivo así como de los sensores de movimientos auxiliares para la detección de movimiento.

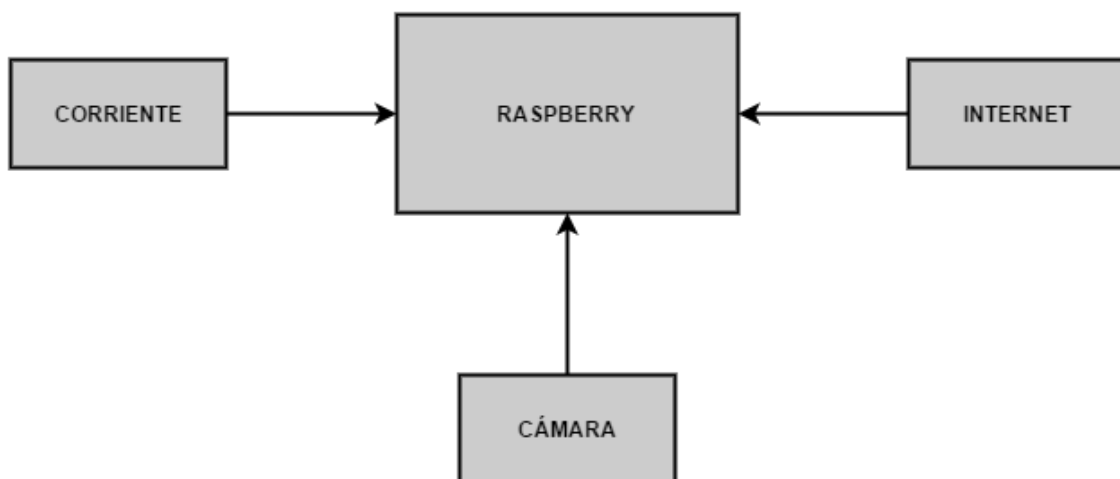


Ilustración 58: Esquema Sistema Videovigilancia Hardware

5.2. Resultado Final del Sistema

En este apartado se procederá a mostrar algunas imágenes mostrando el resultado final del sistema:

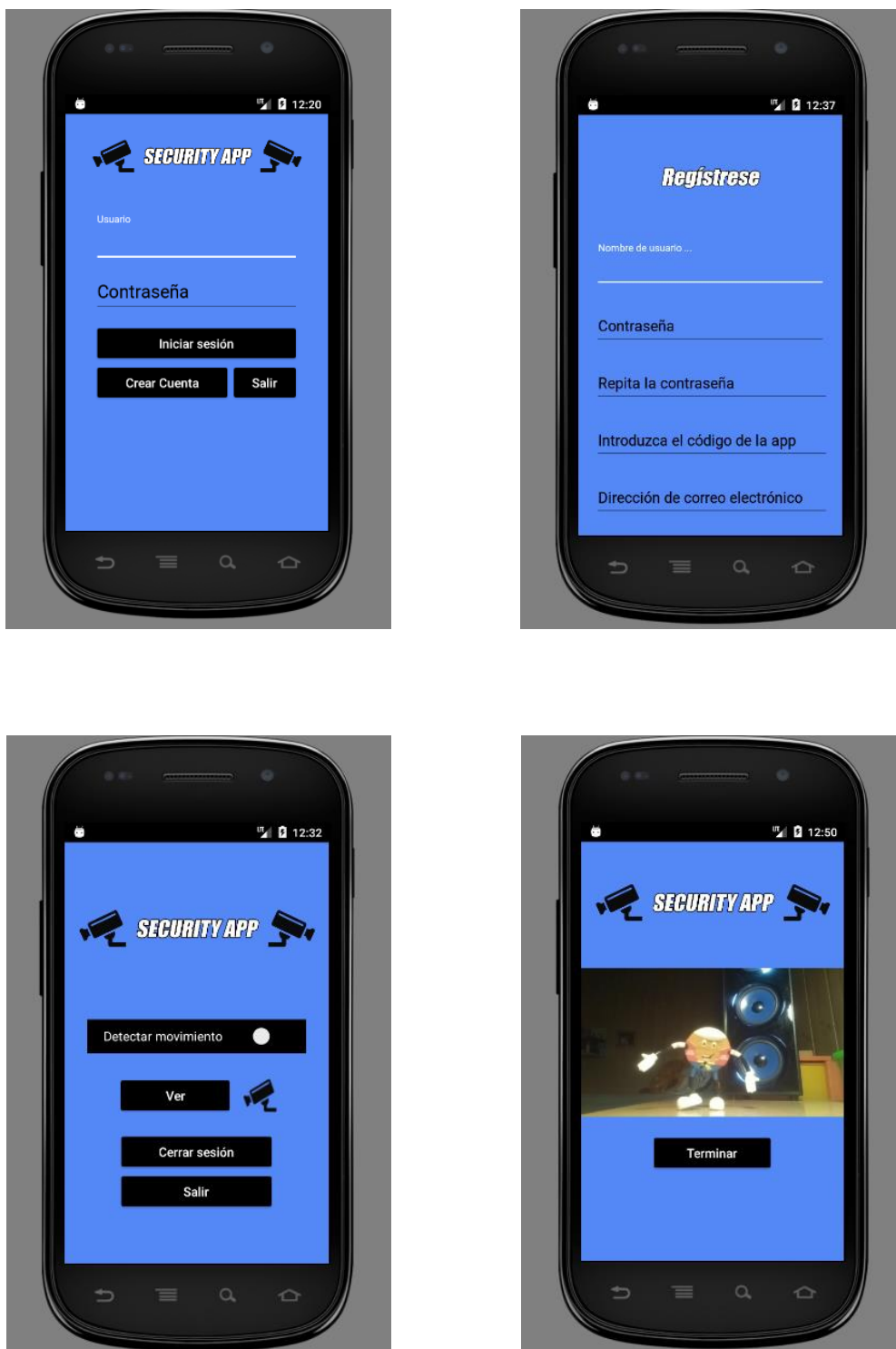


Ilustración 59: Ejemplo visualización aplicación

5.3. Definición del Alcance de las Pruebas

En esta sección se va a proceder a definir qué alcance tienen las pruebas que se van a llevar a cabo, los tipos de pruebas con los que se van a proceder a probar el sistema. Los tipos de pruebas mencionados son las siguientes:

- **Pruebas Unitarias**: Este tipo de pruebas son la batería de pruebas unitarias del sistema, pruebas de caja negra. Estas pruebas se encargarán de probar algunas de las funcionalidades de mayor peso en el sistema de manera individual.
- **Pruebas del Sistema**: Este tipo de pruebas son la batería de pruebas que tratan de testear todo el sistema, las pruebas se realizan sobre todo el sistema completo.

Durante la realización de la prueba, se especificará sobre qué elemento del sistema se está realizando la prueba ya que no se realizarán sobre todos los elementos del sistema.

5.4. Entorno de Pruebas

En esta sección se van a clasificar el entorno sobre el que se van a realizar los diferentes tipos de pruebas, teniendo en cuenta los recursos software y hardware de los equipos utilizados.

- **Equipo 1**: Ordenador de Sobremesa
 - Procesador: 2.7 GHz Intel Core i5 6400
 - Memoria RAM: 8 GB 2133 MHz DDR4 G.Skill Ripjaws V Red
 - Sistema Operativo: Windows 10
- **Equipo 2**: Raspberry Pi 2 Model B
 - Procesador: 900MHz quad-core ARM Cortex-A7 CPU
 - Memoria RAM: 1 GB
 - Sistema Operativo: Raspbian Jessie 8

- **Equipo 3:** BQ Aquaris M5
 - Procesador: 1.5 GHz Qualcomm Snapdragon 615 Octa Core A53
 - Memoria RAM: 2 GB
 - Sistema Operativo: Android Marshmallow 6.0.1
- **Equipo 4:** Android Virtual Device
 - Procesador: 2.7 GHz Intel Core i5 6400 (2 Multi-Core CPU)
 - Memoria RAM: 343 MB
 - Sistema Operativo: Android Marshmallow 6.0

5.5. Especificación de Pruebas

5.5.1. Pruebas Unitarias de Caja Negra

En este apartado se va a proceder a describir las pruebas unitarias de caja negra, las cuales se van a encargar de comprobar que sobre los elementos escogidos para realizar las pruebas, bajo unos parámetros de entrada obtenemos los resultados esperados, cumpliendo así el elemento con su funcionalidad. Dichas pruebas se van a representar con el siguiente formato de tabla:

PUCN-XX	
Título	
Objetivo	
Clase	
Método	
Implementación	
Resultado	

Tabla 61: Formato tabla definición prueba unitaria caja negra

Siendo la definición de los campos que la forman los siguientes:

- **PUCN-XX:** Este campo representa el código que identifica de manera única cada una de las pruebas unitarias de caja negra del sistema. La nomenclatura que se va a proceder a usar es: PUCN-XX, donde XX es el número que representa a la prueba unitaria de caja negra del sistema.

- **Título**: Este campo representa una breve descripción de la prueba.
- **Objetivo**: Este campo representa el objetivo que se busca conseguir con la prueba.
- **Clase**: Este campo representa el nombre de la clase que contiene el elemento de la prueba.
- **Método**: Este campo representa el nombre del método sobre el que se realiza la prueba.
- **Implementación**: Este campo representa los parámetros que se introducen para realizar la prueba.
- **Resultado**: Este campo representa el resultado que se espera con los parámetros introducidos en el campo de implementación.

A continuación se muestran las pruebas unitarias de caja negra del sistema que se han definido:

PUCN-01	
Título	Detectar movimiento.
Objetivo	Detectar movimiento en el sistema de videovigilancia, avisando al usuario con una notificación en caso de detección.
Clase	Servidor
Método	detectorMovimiento
Implementación	Llamar al método.
Resultado	Notificación al Cliente cuando se detecte movimiento.

Tabla 62: Prueba Unitaria Caja Negra PUCN-01

PUCN-02	
Título	Crear un usuario en el sistema.
Objetivo	Insertar el usuario en la base de datos del sistema, devolviendo un valor de resultado.
Clase	Servidor
Método	crearUsuario
Implementación	Introducir los datos del usuario a insertar en la base de datos.
Resultado	<ul style="list-style-type: none"> - Se devuelve el valor 0 si el usuario se ha insertado con éxito. - Se devuelve el valor -1 si el usuario no se ha insertado con éxito.

Tabla 63: Prueba Unitaria Caja Negra PUCN-02

PUCN-03	
Título	Inicio de sesión del usuario
Objetivo	Comprobar si dicho usuario existe en la base de datos y devolver un valor de resultado.
Clase	Servidor
Método	iniciarSesion
Implementación	Introducir el nombre de usuario y la contraseña a buscar y validar en la base de datos.
Resultado	<ul style="list-style-type: none"> - Se devuelve el valor 0 si el usuario existe y la contraseña coincide. - Se devuelve el valor -1 si el usuario no existe o la contraseña no coincide.

Tabla 64: Prueba Unitaria Caja Negra PUCN-03

PUCN-04	
Título	Mandar el vídeo del sistema de videovigilancia al Cliente
Objetivo	Retransmitir en una dirección el contenido multimedia del sistema de videovigilancia
Clase	Servidor
Método	mandarVideo
Implementación	Introducir los datos necesarios de identificación del Cliente
Resultado	<ul style="list-style-type: none"> - Se devuelve el valor 0 si el vídeo se está retransmitiendo con éxito. - Se devuelve el valor -1 si el vídeo no se está retransmitiendo con éxito.

Tabla 65: Prueba Unitaria Caja Negra PUCN-04

PUCN-05	
Título	Activar la detección de movimiento del sistema.
Objetivo	Activar la detección de movimiento en el sistema de videovigilancia.
Clase	Servidor
Método	activarDetector
Implementación	Introducir los datos necesarios de identificación del Cliente
Resultado	<ul style="list-style-type: none"> - Se devuelve el valor 0 si la detección de movimiento se ha activado con éxito. - Se devuelve el valor -1 si la detección de movimiento no se ha activado con éxito.

Tabla 66: Prueba Unitaria Caja Negra PUCN-05

PUCN-06	
Título	Desactivar la detección de movimiento del sistema.
Objetivo	Desactivar la detección de movimiento en el sistema de videovigilancia.
Clase	Servidor
Método	desactivarDetector
Implementación	Introducir los datos necesarios de identificación del Cliente
Resultado	<ul style="list-style-type: none"> - Se devuelve el valor 0 si la detección de movimiento se ha desactivado con éxito. - Se devuelve el valor -1 si la detección de movimiento no se ha desactivado con éxito.

Tabla 67: Prueba Unitaria Caja Negra PUCN-06

PUCN-07	
Título	Crear una notificación de aviso de movimiento
Objetivo	Una vez detectado movimiento, crear una notificación para avisar al usuario de la aplicación.
Clase	Cliente
Método	notificacion
Implementación	Llamar al método
Resultado	Creación y visualización de la notificación al usuario de la aplicación.

Tabla 68: Prueba Unitaria Caja Negra PUCN-07

PUCN-08	
Título	Retransmitir el vídeo del sistema de videovigilancia.
Objetivo	Retransmitir el vídeo del sistema de videovigilancia alojado en una dirección.
Clase	Video
Método	playStream
Implementación	Introducir la dirección donde se encuentra el vídeo.
Resultado	Retransmisión del vídeo procedente de la dirección.

Tabla 69: Prueba Unitaria Caja Negra PUCN-08

5.5.2. Pruebas de Sistema

En este apartado se va a proceder a describir las pruebas de sistema para verificar el correcto funcionamiento del sistema como conjunto, se van a testear funcionalidades que ofrece la interfaz de usuario. Para definir este tipo de pruebas se va a realizar con el siguiente formato de tabla:

PS-XX	
Nombre	
Objetivo	
Implementación	
Resultado	

Tabla 70: Formato tabla definición prueba sistema

Siendo la definición de los campos que la forman los siguientes:

- **PS-XX**: Este campo representa el código que identifica de manera única cada una de las pruebas de sistema. La nomenclatura que se va a proceder a usar es: PS-XX, donde XX es el número que representa a la prueba del sistema.
- **Nombre**: Este campo representa una breve descripción de la prueba.
- **Objetivo**: Este campo representa el objetivo que se busca conseguir con la prueba.
- **Implementación**: Este campo representa los pasos a seguir para realizar la prueba.
- **Resultado**: Este campo representa el resultado que se espera tras realizar los pasos de la implementación.

A continuación se muestran las pruebas de sistema que se han definido:

PS-01	
Nombre	Acceso a la aplicación
Objetivo	Demostrar que al acceder a la aplicación sin iniciar sesión anteriormente, se accede a la página de Login y se muestran las opciones de iniciar sesión, crear cuenta y salir de la aplicación
Implementación	1. Abrir la aplicación. 2. Observar el resultado obtenido.
Resultado	Se muestra la sección de Login al usuario.

Tabla 71: Prueba Sistema PS-01

PS-02	
Nombre	Crear cuenta
Objetivo	Demostrar que al pulsar en la opción de Crear Cuenta, se nos redirige a la sección de crear cuenta y nos muestra el formulario de registro, pudiendo enviar los datos si están validados y devolviéndonos a la sección Login si está todo correcto.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Pulsar en Crear cuenta. 3. Observar el formulario de registro. 4. Rellenar formulario de registro. 5. Enviar formulario de registro. 6. Observar que se está en la sección de Login.
Resultado	Se muestra al usuario un mensaje con la confirmación de la creación del usuario, redirigiéndole al Login.

Tabla 72: Prueba Sistema PS-02

PS-03	
Nombre	Iniciar sesión
Objetivo	Demostrar que al introducir el nombre de usuario y la contraseña de un usuario creado anteriormente, permite al usuario acceder a la sección de Inicio.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Rellenar los campos de usuario y contraseña. 3. Pulsar en Iniciar sesión. 4. Observar que se está en la sección de Inicio.
Resultado	Se muestra al usuario un mensaje con la confirmación del inicio de sesión, redirigiéndole a Inicio.

Tabla 73: Prueba Sistema PS-03

PS-04	
Nombre	Cerrar sesión
Objetivo	Demostrar que al pulsar en la opción de Cerrar sesión, se nos redirige a la sección de Login.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Rellenar los campos de usuario y contraseña. 3. Pulsar en Iniciar sesión. 4. Observar que se está en la sección de Inicio. 5. Pulsar en Cerrar sesión. 6. Observar que se está en la sección de Login.
Resultado	Se muestra al usuario un mensaje con la confirmación del cierre de sesión, redirigiéndole al Login.

Tabla 74: Prueba Sistema PS-04

PS-05	
Nombre	Salir de la aplicación
Objetivo	Demostrar que al pulsar en la opción de Salir, salimos de la aplicación sin cerrar la sesión del usuario.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Rellenar los campos de usuario y contraseña. 3. Pulsar en Iniciar sesión. 4. Observar que se está en la sección de Inicio. 5. Pulsar en Salir. 6. Observar que se ha cerrado la aplicación.
Resultado	Se cierra la aplicación manteniendo la sesión iniciada.

Tabla 75: Prueba Sistema PS-05

PS-06	
Nombre	Acceso a la aplicación con sesión iniciada.
Objetivo	Demostrar que al salir de la aplicación con la sesión iniciada, al volver a entrar la sesión sigue iniciada.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Rellenar los campos de usuario y contraseña. 3. Pulsar en Iniciar sesión. 4. Observar que se está en la sección de Inicio. 5. Pulsar en Salir. 6. Observar que se ha cerrado la aplicación. 7. Abrir la aplicación. 8. Observar que nos encontramos en la sección de Inicio.
Resultado	Se redirige al usuario a la sección Inicio.

Tabla 76: Prueba Sistema PS-06

PS-07	
Nombre	Activar detector de movimiento.
Objetivo	Demostrar que se le permite al usuario activar la detección de movimiento y avisarle con una notificación en caso de detección.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Rellenar los campos de usuario y contraseña. 3. Pulsar en Iniciar sesión. 4. Observar que se está en la sección de Inicio. 5. Pulsar en Detectar movimiento. 6. Observar que se confirma la activación con un mensaje.
Resultado	Activar la detección de movimiento del sistema de videovigilancia.

Tabla 77: Prueba Sistema PS-07

PS-08	
Nombre	Desactivar detector de movimiento.
Objetivo	Demostrar que se le permite al usuario desactivar la detección de movimiento.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Rellenar los campos de usuario y contraseña. 3. Pulsar en Iniciar sesión. 4. Observar que se está en la sección de Inicio. 5. Pulsar en Detectar movimiento. 6. Observar que se confirma la desactivación con un mensaje.
Resultado	Desactivar la detección de movimiento del sistema de videovigilancia.

Tabla 78: Prueba Sistema PS-08

PS-09	
Nombre	Ver video
Objetivo	Demostrar que se le permite al usuario ver el video del sistema de videovigilancia.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Rellenar los campos de usuario y contraseña. 3. Pulsar en Iniciar sesión. 4. Observar que se está en la sección de Inicio. 5. Pulsar en Ver. 6. Observar que se está en la sección de Vídeo. 7. Observar que se muestra la retransmisión del video del sistema de videovigilancia.
Resultado	Mostrar al usuario la retransmisión en vivo del sistema de videovigilancia.

Tabla 79: Prueba Sistema PS-09

PS-10	
Nombre	Terminar de ver video
Objetivo	Demostrar que se le permite al usuario dejar de ver el video del sistema de videovigilancia.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Rellenar los campos de usuario y contraseña. 3. Pulsar en Iniciar sesión. 4. Observar que se está en la sección de Inicio. 5. Pulsar en Ver. 6. Observar que se está en la sección de Vídeo. 7. Observar que se muestra la retransmisión del video del sistema de videovigilancia. 8. Pulsar en Terminar. 9. Observar que se está en la sección de Inicio.
Resultado	Dejar de mostrar al usuario la retransmisión en vivo del sistema de videovigilancia.

Tabla 80: Prueba Sistema PS-10

PS-11	
Nombre	Notificación movimiento
Objetivo	Demostrar que se avisa al usuario con una notificación en caso de detectar movimiento.
Implementación	<ol style="list-style-type: none"> 1. Abrir la aplicación. 2. Rellenar los campos de usuario y contraseña. 3. Pulsar en Iniciar sesión. 4. Observar que se está en la sección de Inicio. 5. Provocar movimiento en el sistema de videovigilancia. 6. Observar que se recibe una notificación avisando del movimiento. 7. Observar que se permite ver el video a través de la notificación recibida. 8. Observar el vídeo del sistema de videovigilancia.
Resultado	Crear una notificación para el usuario cuando se detecte movimiento, así como darle posibilidad de ver el vídeo.

Tabla 81: Prueba Sistema PS-11

PS-12	
Nombre	Obtener estado del sistema
Objetivo	Demostrar que se le permite al administrador obtener el estado del sistema.
Implementación	<ol style="list-style-type: none"> 1. Entrar en el sistema como administrador. 2. Solicitar el estado actual del sistema. 3. Observar que se recibe el estado actual del sistema.
Resultado	Recibir el estado actual del sistema así como las incidencias ocurridas.

Tabla 82: Prueba Sistema PS-12

5.6. Análisis de Consistencia

En este apartado se mostrará la forma en que las pruebas definidas cubren de forma correcta los requisitos de software del sistema, definidos en el apartado 3 de Análisis del sistema.

Para mostrarlo se hará uso de una matriz de trazabilidad Requisitos/Pruebas para posteriormente comprobar de manera visual que la funcionalidad del sistema se verifica con las pruebas que se han definido anteriormente.

	PUCN-01	PUCN-02	PUCN-03	PUCN-04	PUCN-05	PUCN-06	PUCN-07	PUCN-08	PS-01	PS-02	PS-03	PS-04	PS-05	PS-06	PS-07	PS-08	PS-09	PS-10	PS-11	PS-12
RF-01	X	X	X	X			X	X		X	X	X	X	X	X	X	X	X	X	X
RF-02		X	X							X	X	X	X	X	X	X	X	X	X	X
RF-03		X								X										
RF-04		X								X										
RF-05			X								X	X	X	X	X	X	X	X	X	
RF-06	X			X				X									X	X	X	
RF-07				X				X									X	X	X	
RF-08																		X		
RF-09	X				X	X	X												X	
RF-10					X										X				X	
RF-11						X										X				
RF-12	X						X												X	
RF-13		X								X										
RF-14		X								X										
RF-15			X								X	X	X	X	X	X	X	X	X	
RF-16												X								
RF-17													X	X						
RF-18	X				X	X	X													
RF-19	X																			X
RF-20	X																			X
RNF-01		X	X		X	X		X		X	X	X	X	X	X	X	X	X	X	
RNF-02		X	X		X	X		X		X	X	X	X	X	X	X	X	X	X	
RNF-03		X	X		X	X		X		X	X	X	X	X	X	X	X	X	X	
RNF-04																				X
RNF-05			X							X	X	X	X	X	X	X	X	X	X	
RNF-06					X	X			X		X	X	X	X	X	X	X	X	X	
RNF-07								X									X	X	X	
RNF-08		X								X										

Tabla 83: Matriz trazabilidad Requisitos/Pruebas

6. Gestión del Proyecto

En este capítulo se va a proceder a detallar tanto la planificación como la gestión del proyecto que se ha llevado a cabo durante todo el periodo de su realización, haciendo una comparativa entre los tiempos que se habían estimado con los tiempos reales y a su vez finales. Se añadirá una estimación del presupuesto del proyecto.

6.1. Planificación Temporal

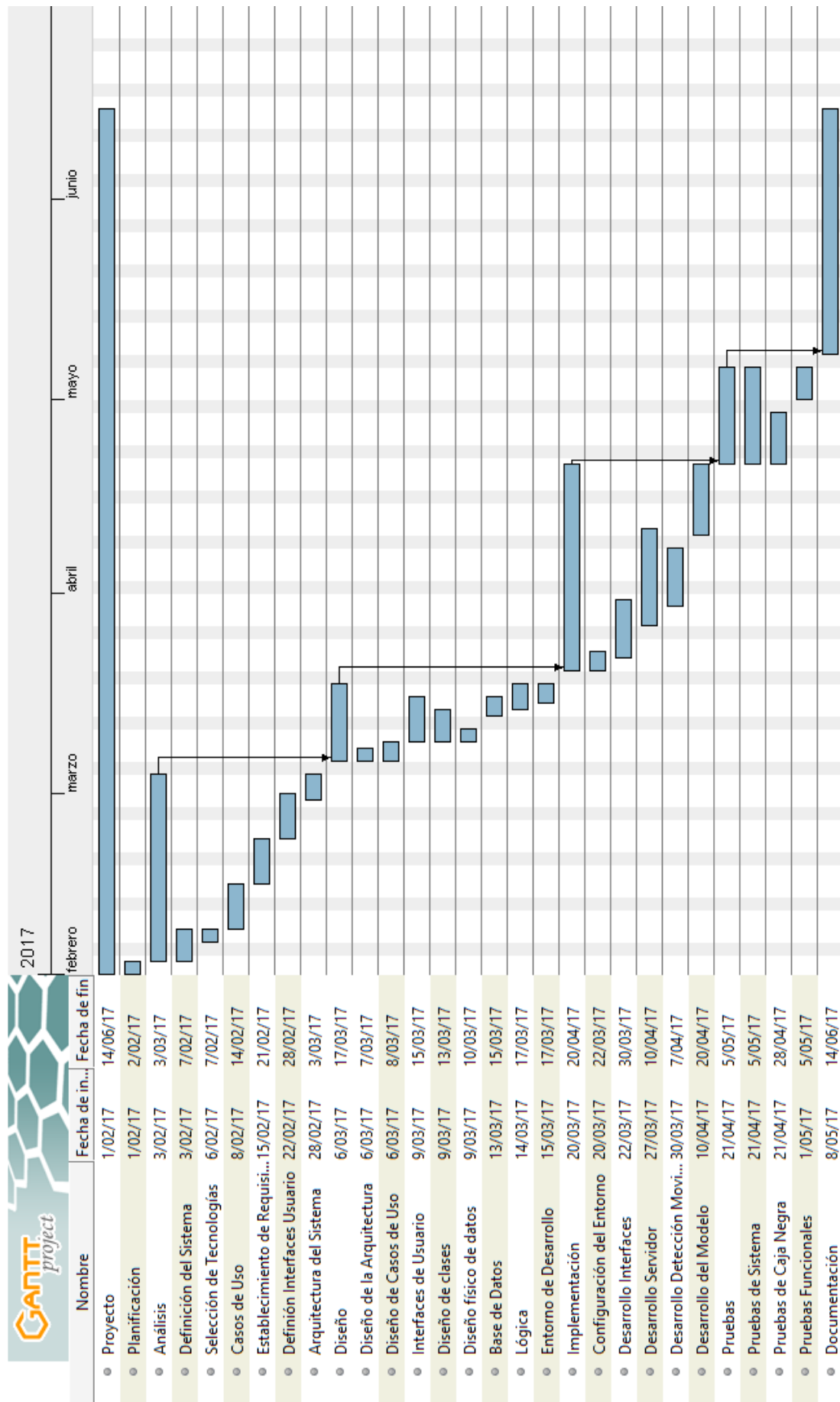
En esta sección se va a proceder a exponer tanto la gestión como la planificación que se ha llevado a cabo, haciendo una comparativa entre la planificación que se realizó al principio con la planificación real que se ha llevado a cabo para realizar el proyecto finalmente.

6.1.1. Planificación Inicial

En este apartado se va a proceder a definir la planificación inicial del proyecto. La fecha de inicio del proyecto se establece el día 01 de Febrero de 2017, siendo la fecha de fin de proyecto establecida el día 04 de Junio de 2017, aproximadamente unos 4 meses de duración.

El esfuerzo diario invertido se estableció en 4 horas, siendo 1 hora dedicada al descanso (1.5 horas, 1 hora, 1.5 horas) dedicada de lunes a viernes sin contar los fines de semana.

A continuación se muestra la planificación inicial mencionada en este apartado:

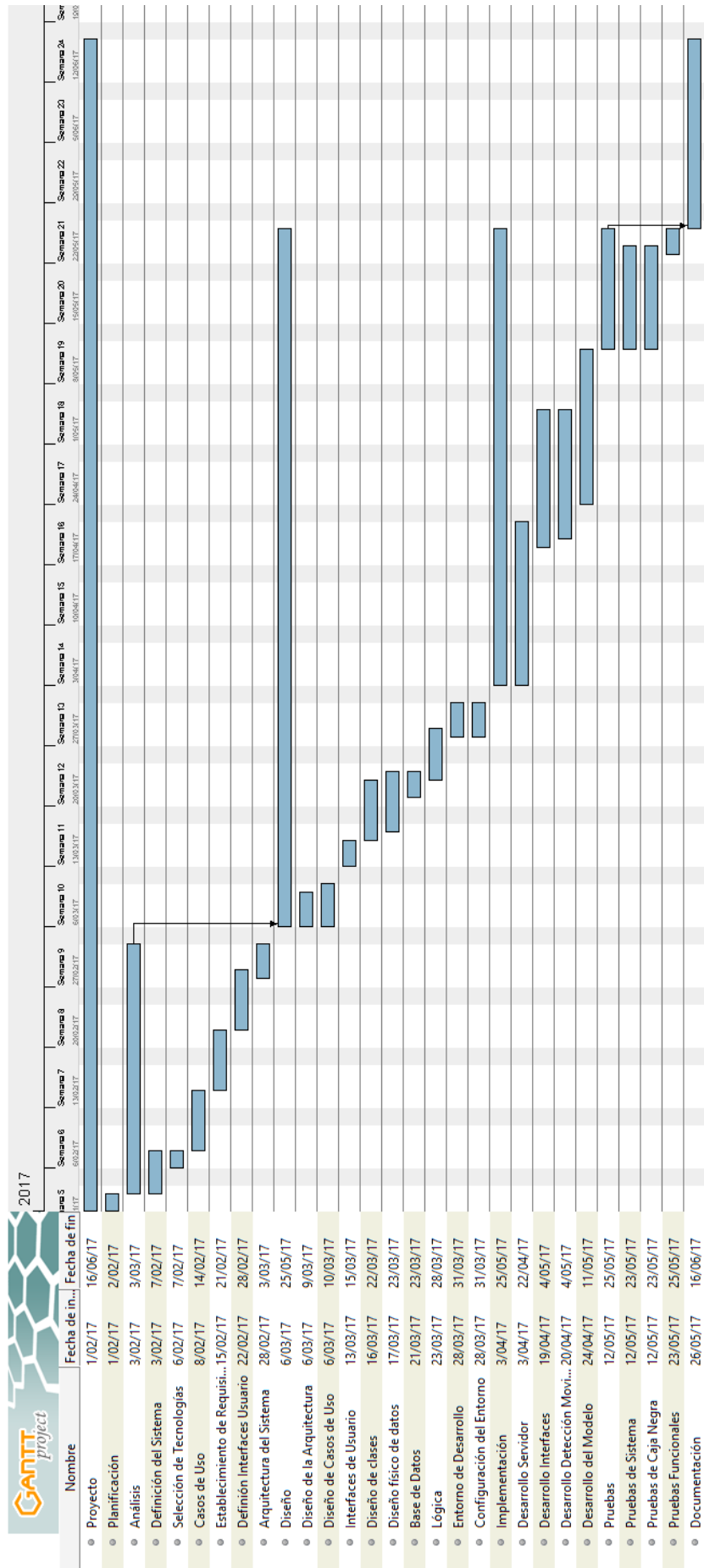


6.1.2. Desarrollo Real del Proyecto

En el apartado anterior se ha mostrado la planificación inicial del proyecto, sin embargo la duración de las tareas se han visto afectadas en comparación a lo estimado anteriormente. Todo ello es debido a los exámenes, las prácticas que se han debido de realizar durante este periodo así como los exámenes finales de mayo, teniendo en cuenta además la poca experiencia que se tiene a la hora de estimar las tareas de un proyecto.

En este apartado se procederá a mostrar la planificación del proyecto real que se ha llevado a cabo para completar todas las tareas requeridas, sufriendo un ligero retraso en la entrega final ya que la entrega se realizó el día 16 de Junio de 2017, pero en líneas generales manteniendo la estimación inicial.

Se han encontrado cierto tipo de dificultades no esperadas a la hora de implementar las diferentes tareas del proyecto, así como ha habido tareas que se han resuelto en menor tiempo. Durante el intervalo que dura la tarea no significa que durante todo ese intervalo se haya estado trabajando en el proyecto, en más de un caso dentro del intervalo de días existirán días de descanso, se ha establecido la fecha de cuando se empezó la tarea y la fecha cuando se terminó. La planificación real del proyecto es la siguiente:



Como se puede observar a simple vista la tarea de Diseño y la tarea de Implementación se alargan hasta el final de las pruebas, esto es debido a que a medida que se ha ido avanzando en el proyecto, se ha visto necesario pasar a algún punto anterior para realizar pequeñas modificaciones que a priori no se habían tenido en cuenta.

6.2. Presupuesto

En este apartado se va a proceder a describir y definir el análisis económico del presupuesto de una manera estimada, en función de la duración real que ha tenido el Trabajo Fin de Grado. Para llevarlo a cabo se presentará en primer lugar el coste total que ha supuesto el proyecto desglosando los costes en los posteriores apartados.

6.2.1. Presupuesto Total

El proyecto con título: *Diseño e Implementación de un Sistema de Control Remoto para Vigilancia en Android*. Con una duración total de **4 meses y 15 días** (con la inclusión de días no laborables).

Tiene un presupuesto total para su realización de **26,343.63 (VEINTISEIS MIL TRESCIENTOS CUARENTA Y TRES CON SESENTA Y TRES EUROS)** (I.V.A. incluido).

En los siguientes apartados se procederá a realizar el desglose correspondiente del coste total del proyecto justificando cada uno de los costes.

6.2.2. Desglose del Presupuesto

En este apartado se va a proceder a describir cada uno de los gastos que se han tenido en cuenta para calcular el presupuesto total del proyecto.

6.2.2.1. Costes directos

Los costes directos del proyecto son los siguientes:

- **Costes de Personal:** para el desarrollo del proyecto se han tenido que realizar diversas tareas que habitualmente se realizan por distintos roles dentro del proyecto [20]. Dichos roles son los siguientes:
 - *Jefe de proyecto:* Este miembro del proyecto es el encargado de realizar la supervisión del trabajo del resto de personas así como asegurarse de que el proyecto cumple con los objetivos que se han marcado.
 - *Analista:* Este miembro del proyecto es el encargado de analizar las necesidades que tiene el usuario y mostrarlas en los casos de uso, en los requisitos de software, la elección de las tecnologías a usar, etc.
 - *Diseñador:* Este miembro del proyecto es el encargado de diseñar el sistema teniendo en cuenta el análisis previamente realizado por el analista. Será el encargado de diseñar las clases, el modelo físico de datos, las interfaces, etc.
 - *Programador:* Este miembro del proyecto es el encargado de implementar lo diseñado anteriormente por el diseñador.
 - *Gestor de Pruebas:* Este miembro del proyecto es el encargado de verificar que lo implementado por el programador cumple las funcionalidades requeridas por el cliente y funciona correctamente.

Los roles definidos realizan tareas distintas dentro del proyecto, por lo que sus salarios serán distintos:

Rol	Horas empleadas (h)	Coste (€/h)	Coste
Jefe de Proyecto	120	38.05	4,566.00 €
Analista	55	26.04	1,432.20 €
Diseñador	104	25.24	2,624.96 €
Programador	333	21.74	7,239.42 €
Gestor de Pruebas	74	18.64	1,379.36 €
TOTAL	686	-	17,241.94 €

Tabla 84: Costes Personal

- **Costes de Material:** para realizar el proyecto también se tiene que llevar a cabo una serie de gastos en material informático, tanto hardware como software así

como material fungible. Los equipos que se han empleado para el desarrollo del proyecto eran de la propiedad del autor, por lo que la amortización se realizará teniendo en cuenta la duración del proyecto y suponiendo el caso de que el equipo informático se amortizará de manera completa en un periodo de 3 años.

Unidades	Componente	Coste	Meses de uso	Coste aplicable al proyecto
1	Ordenador sobremesa	700.00 €	4	175.00 €
1	Raspberry Pi 2 Model B	25.00 €	4	6.25 €
1	Módulo Cámara Raspberry Pi 2 Model B	26.00 €	4	6.50 €
1	LG Smart TV 32"	280.00 €	4	70.00 €
1	Monitor Hacer 21"	100.00 €	4	25.00 €
1	BQ Aquaris M5 16 GB	180.00 €	4	45.00 €
1	Licencia Windows 10	280.00 €	4	70.00 €
1	Microsoft Office 2016	150.00 €	4	37.50 €
TOTAL		435.25 €		

Tabla 85: Costes material informático

Unidades	Artículo	Coste unidad	Coste
5	Paquete 500 folios Din A4 80gr	3.75 €	18.75 €
5	Pack 50 bolígrafos BIC	15.00 €	75.00 €
5	Blocs Notas adhesivas 76x76 mm	7.00 €	35.00 €
TOTAL		128.75 €	

Tabla 86: Costes material fungible

6.2.2.2. Costes Indirectos

Los costes indirectos que están asociados al proyecto se van a proceder a calcular con una estimación en forma de porcentaje sobre los costes directos del proyecto, siendo el porcentaje de un 6%. Se incluyen dentro de estos gastos los causados por la luz, el agua, la conexión a internet, el teléfono, etc.

Costes indirectos	
6% sobre gastos Directos	1,034.52 €

Tabla 87: Costes indirectos

Para calcular el beneficio a conseguir por la realización del presente proyecto, se ha procedido a analizar estudios y datos que se han conseguido para el mismo. Dicho estudio ha dado como resultado un beneficio del 17% sobre los costes directos del proyecto.

Beneficios	
17% sobre gastos Directos	2,931.13 €

Tabla 88: Beneficios

7.2.3. Resumen de Costes

En este apartado se mostrarán los costes totales del proyecto sin incluir el IVA y los costes totales del proyecto incluyendo el IVA, agrupando los distintos costes que se han ido mencionando en los apartados anteriores:

Resumen de Costes	
Costes Directos	
Salarios	17,241.94 €
Material informático	435.25 €
Material fungible	128.75 €
Costes indirectos	1,034.52 €
Beneficios	2,931.13 €
TOTAL sin IVA	21,771.59 €
IVA (21%)	4,572.04 €
TOTAL con IVA	26,343.63 €

Tabla 89: Resumen costes

7. Marco Regulador y Ético

En este apartado se va a proceder a describir el aspecto legal de todo aquello referente a este proyecto. Se van a detallar los diversos estándares así como las licencias de las tecnologías utilizadas en el proyecto.

7.1. Android Studio

El lenguaje de programación que se ha utilizado para desarrollar la parte del subsistema Cliente ha sido Java, sin embargo dicho lenguaje de programación viene incluido en Android Studio, por lo que nos fijaremos en los aspectos legales de este.

Los términos y condiciones de este entorno de desarrollo son expuestos en su página web [32], básicamente concediéndonos la licencia del software siempre y cuando los términos y condiciones se acepten ya que se trata de un tipo de software abierto y libre proporcionado por Google.

7.2. Geany

El lenguaje de programación C# que se ha utilizado para desarrollar la parte del subsistema Servidor es de licencia libre, sin embargo los entornos de desarrollo que se suelen utilizar para programar en este lenguaje no. En este caso se ha utilizado el entorno de desarrollo Geany el cual utiliza la GLP de GNU [33] la cuál es de las más usadas en el ámbito de software libre y código abierto, garantizando al usuario final la libertad de usar, estudiar, compartir y modificar el software.

7.3. Python

Este lenguaje de programación se ha utilizado para desarrollar parte del subsistema Servidor, el cual posee una licencia pero a día de hoy está en desuso [34]. Se trata de una licencia de software libre sin embargo no es compatible con la Licencia Pública General de GNU mencionada en el apartado anterior, es por ello que dicha licencia se retiró siendo este lenguaje de programación software libre. Se ha desarrollado con el entorno de desarrollo Geany.

7.4. MySQL

Este gestor de bases de datos relacional utilizado para la base de datos de los usuarios de la aplicación del sistema, está desarrollado bajo la licencia dual Licencia Pública General/Licencia comercial por Oracle Corporation, considerada la base de datos de código abierto más popular del mundo [35].

7.5. VLC

VLC media player es un reproductor y framework multimedia de software y código libre bajo la Licencia Pública General, desarrollado por el proyecto VideoLAN. En el proyecto se ha utilizado para retransmitir en streaming el contenido del sistema de videovigilancia a una dirección para que el Cliente pudiera visualizarla en su dispositivo Android [36].

8. Conclusiones y Trabajos Futuros

En este apartado se mostrarán las conclusiones que se han extraído al realizar el presente Trabajo Fin de Grado, planteando posibles líneas de desarrollo futuras para mejorar la funcionalidad del sistema.

8.1. Conclusiones

Antes de empezar el presente Trabajo Fin de Grado se tuvo en cuenta que el mundo del sistema de la videovigilancia está explotado, por lo que se debía innovar en el aspecto de la cercanía y sencillez además de precisión y seguridad, intentar mejorar lo existente.

La funcionalidad principal de videovigilancia con detección de movimiento planteada al principio del Trabajo Fin de Grado se ha conseguido de manera exitosa, ya que se disponen de multitud de herramientas, librerías y entornos de desarrollo para ello, lo complicado es amoldar y encajar dichas herramientas para conseguir tus objetivos propuestos en cuanto a funcionalidad se refiere.

Los sistemas de videovigilancias actuales constan de cámaras y sensores de gran precisión pero también caros. Lo que se ha perseguido en este Trabajo Fin de Grado es realizar las mismas funcionalidades o de maneras muy similares pero de una forma mucho más económica, ya que no es lo mismo usar en el sistema cámaras de videovigilancia potentes que una Raspberry Pi, sin embargo en este Trabajo Fin de Grado se demuestra que pueden ofrecer resultados similares, por lo que puede ser una idea interesante de cara a la videovigilancia además de su adaptación a la plataforma Android.

Dejando un poco de lado los sistemas de videovigilancias actuales puros y duros, también se quiere hacer conciencia en que puedes tener en casa o en tu establecimiento un sistema que te avise ante cierto evento que tú le instales previamente, de una forma muy económica y sobre todo segura, no solo se cierra al ámbito de grabar si no que puede configurarse para cualquier evento, por ello es una idea interesante y sobre todo mucho más económica.

8.2. Conclusiones personales

En cuanto a la conclusión personal obtenida tras la realización del Trabajo Fin de Grado, siempre ha sido algo que se ha visto desde lejos como algo que nunca llegaría pero aquí estoy escribiendo las conclusiones que he obtenido al realizarlo, muy contento con lo que he desarrollado en este proyecto así como de las cosas nuevas que he aprendido y he reforzado.

Al principio no sabía si iba a ser capaz de al menos desarrollar algo decente, pero empiezas poco a poco a buscar información, a probar herramientas, a preguntar sugerencias sobre cómo empezar, y paulatinamente el proyecto iba cogiendo forma y cada vez me ponía con ello con más ganas, a medida que iba observando que lo iba terminando, que el proyecto iba cogiendo forma.

Como he mencionado anteriormente, con la realización del presente Trabajo Fin de Grado he asentado los conocimientos adquiridos durante estos 4 años en la carrera así como he continuado aprendiendo conocimientos nuevos, por lo que esa sensación de haber desarrollado una aplicación que funciona y cumple la funcionalidad requerida y encima se han aprendido conocimientos nuevos, pues estoy contento con todos estos resultados obtenidos, es algo que veía lejos hace años pero que a día de hoy ya está terminado, una etapa más de mi vida finalizada pero esto no ha hecho nada más que comenzar.

8.3. Trabajos Futuros

Cualquier aspecto de la vida es mejorable, un proyecto software no es menos. Se pueden añadir mejoras al proyecto desarrollado de manera que mejoren u optimicen las funcionalidades implementadas así como añadir nuevas funcionalidades, de hecho el proyecto está desarrollado de forma que sea fácil y sencillo tanto añadir como quitar funcionalidades. Algunas de las posibles mejoras son las siguientes:

1. Se puede añadir la funcionalidad de reconocimiento facial al sistema junto a la funcionalidad de detección de movimiento. Bastaría con añadir una nueva base de datos con las imágenes faciales a reconocer e implantar dicha funcionalidad en el sistema.
2. Mejorar la interfaz de usuario de la aplicación. Las interfaces en su gran mayoría se pueden mejorar, se ha pretendido usar una interfaz que no asuste al usuario pero es uno de los aspectos de mejora sin duda de la aplicación del sistema.
3. Aunque el sistema almacena el evento del movimiento mediante imágenes (ya que la Raspberry no tiene mucha memoria para almacenar videos), se podría añadir la funcionalidad de mostrar dichas imágenes al usuario así como darle la oportunidad de visualizar el historial de eventos y ver el contenido de cada uno de ellos. Para ello se necesitaría una memoria más grande pero la funcionalidad se podría añadir de manera sencilla.

9. Referencias

- [1] I. F. Lantigua, «El móvil supera por primera vez al ordenador para acceder a Internet», *El Mundo*, 04 Abril 2016.
- [2] P. Villanueva, «El número de usuarios de Internet en el mundo alcanza el 50% de la población (2017) », *Marketing Ecommerce*, 09 Febrero 2017.
- [3] Wikipedia, «Videovigilancia IP», *Wikipedia*, 10 Abril 2017.
- [4] Geany, «About Geany», *Geany*, 15 Febrero 2010 [En línea]. Available: <http://www.geany.org/Main/About>. [Último acceso: 06 Junio 2017]
- [5] Developers Android, «Conoce Android Studio», *Developers Android* [En línea]. Available: <https://developer.android.com/studio/intro/index.html?hl=es-419>. [Último acceso: 06 Junio 2017].
- [6] Videovigilancia, «¿Qué es la videovigilancia?», *Videovigilancia* [En línea]. Available: <http://videovigilancia.com/respvideovigilancia.htm> [Último acceso: 08 Junio 2017].
- [7] Prevent Security Systems, «Seguridad y videovigilancia con Drones», *Prevent Security Systems* [En línea]. Available: <https://www.prevent.es/servicios-de-seguridad/camaras-de-seguridad/empresas/seguridad-y-videovigilancia-con-drones> [Último acceso: 08 Junio 2017].
- [8] Alarmas y videovigilancia, «¿Detección de movimiento o entrada de alarma? », *Alarmas y videovigilancia*, 05 Mayo 2014 [En línea]. Available: <http://alarmasyvideovigilancia.com/deteccion-de-movimiento-o-entrada-de-alarma/> [Último acceso: 08 Junio 2017].
- [9] Axis Communications, «Reconocimiento facial», *Axis Communications* [En línea]. Available: <https://www.axis.com/es/es/solutions-by-application/facial-recognition> [Último acceso: 08 Junio 2017].
- [10] Securitas Direct, «Historia de Securitas Direct: la empresa de las alarmas», *Securitas Direct* [En línea]. Available: <https://www.securitasdirect.es/es/empresa/historia> [Último acceso: 19 Junio 2017].
- [11] Securitas Direct, «Videovigilancia», *Securitas Direct* [En línea]. Available: <https://www.securitasdirect.es/es/videovigilancia> [Último acceso: 19 Junio 2017].

- [12] IMSEL Seguridad S.A., «IMSEL Seguridad S.A. ¡Tu mejor elección!», *ImSEL Seguridad S.A.* [En línea] Available: <http://www.imsel.com/empresas-seguridad-privada/> [Último acceso: 19 Junio 2017]
- [13] IMSEL Seguridad S.A., «Cámaras y sistemas de Videovigilancia», *IMSEL Seguridad S.A.* [En línea] Available: <http://www.imsel.com/camaras-de-videovigilancia/> [Último acceso: 19 Junio 2017].
- [14] Altatec Seguridad, «Quiénes somos, *Altatec Seguridad*» [En línea] Available: <http://www.altatec-seguridad.com/quienes-somos/> [Último acceso: 19 Junio 2017].
- [15] Altatec Seguridad, «Videovigilancia en empresas», *Altatec Seguridad* [En línea] Available: <http://www.altatec-seguridad.com/camaras-de-seguridad-para-empresas/> [Último acceso: 19 Junio 2017].
- [16] Arduino, «¿Qué es Arduino? », *Arduino* [En línea] Available: <http://arduino.cl/que-es-arduino/> [Último acceso: 19 Junio 2017].
- [17] Prometec, «Arduino Cámara VGA», *Prometec* [En línea] Available: <https://www.prometec.net/producto/arduino-camara-vga/> [Último acceso: 19 Junio 2017].
- [18] Electronilab, «Sensor de movimiento PIR HC-SR501», *Electronilab* [En línea] Available: <https://electronilab.co/tienda/sensor-de-movimiento-pir-hc-sr501/> [Último acceso: 19 Junio 2017].
- [19] A. Castro, «¿Qué es Raspberry Pi, dónde comprarla y cómo usarla? », *Computer Hoy*, 23 Enero 2017.
- [20] Electronilab, «Cámara para Raspberry Pi 5MP», *Electronilab* [En línea] Available: <https://electronilab.co/tienda/camara-para-raspberry-pi-5mp/> [Último acceso: 19 Junio 2017].
- [21] Redeweb, «Métodos de detección de movimiento. Comparativa de la tecnología de los sensores PIR de Nicera, WaveEye de NJRC y GridEye de Panasonic», *Redeweb* [En línea] Available: <http://www.redeweb.com/articulos/articulo.php?inicio=0&pag=1&id=1380> [Último acceso: 19 Junio 2017].
- [22] EcuRed, «Arquitectura Cliente Servidor», *EcuRed* [En línea]. Available: https://www.ecured.cu/Arquitectura_Cliente_Servidor [Último acceso: 11 Junio 2017].

- [23] Java, «¿Qué es la tecnología Java y para qué la necesito? », *Java* [En línea]. Available: https://www.java.com/es/download/faq/whatis_java.xml [Último acceso: 13 Junio 2017].
- [24] EcuRed, «Lenguaje de Programación C», *EcuRed* [En línea]. Available: https://www.java.com/es/download/faq/whatis_java.xml [Último acceso: 13 Junio 2017].
- [25] M.A. Álvarez, «Qué es Python», *Desarrolloweb*, 19 Noviembre 2003.
- [26] W. Sagástegui, «¿Qué es y para qué sirve el lenguaje de etiquetas XML (Extensible Markup Language)? », *Aprenderaprogramar* [En línea]. Available: http://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=102:ique-es-y-para-que-sirve-el-lenguaje-de-etiquetas-xml-extensible-markup-language&catid=46&Itemid=163 [Último acceso: 13 Junio 2017].
- [27] J. Pérez, A. Gardey, «Definición de XML», *Definicion* [En línea]. Available: <http://definicion.de/xml/> [Último acceso: 13 Junio 2017].
- [28] Academia Android, «Android Studio v1.0: características y comparativa con Eclipse», *Academia Android*, 11 Diciembre 2014.
- [29] EcuRed, «Geany», *EcuRed* [En línea]. Available: <https://www.ecured.cu/Geany> [Último acceso: 13 Junio 2017].
- [30] ISocialWeb, «MYSQL ¿Qué es y para qué sirve? », *ISocialWeb*, 26 Agosto 2014.
- [31] Instituto Superior de Formación y Recursos en Red para el Profesorado, «¿Qué es VLC Media Player? », *Instituto Superior de Formación y Recursos en Red para el Profesorado* [En línea]. Available: <http://www.ite.educacion.es/formacion/materiales/107/cd/video/video0201.html> [Último acceso: 14 Junio 2017].
- [32] HAYS, Guía del mercado laboral 2017, *HAYS*, Barcelona, 2017
- [33] Developer Android, Términos y condiciones, *Developer Android*, 09 Diciembre 2016 [En línea]. Available: <https://developer.android.com/studio/terms.html?hl=es-419> [Último acceso: 16 Junio 2017]
- [34] Wikipedia, Geany, *Wikipedia*, 14 Abril 2017.
- [35] Wikipedia Móvil, Python, *Wikipedia Móvil*, 05 Abril 2016.
- [36] Wikipedia, MySQL, *Wikipedia*, 30 Mayo 2017.
- [37] Wikipedia, VLC, *Wikipedia*, 15 Junio 2017.

Anexo: Summary

Abstract

This End of Grade Work objective is developing an application in Android, so the user can look out her/his house or the place she/he has chosen to look out. It is looked for closeness and simplicity to the user besides security and protection.

This application will let the user to see live videos from videocameras installed as well as activate movement detection system, so in case of detecting some kind of movement the user will be warned by an Android notification in her/his device which the application is installed.

To develop this application it is used Server-Client model where Client is the Android user device and the Server is the mechanism which will control cameras movement and it also will warn the user in case of some kind of movement. Server will be composed by tools to control the correct functioning and the correct user and videocameras synchrony as well as the movement warnings.

The final objective is the implementation of this application in Android platform and the implementation of this Server to attend all kind of requests from the different users of the Android's application.

Introduction

Nowadays mobile devices are the most used compared with computers, televisions or video games consoles [1]. That is the reason because nowadays applications and tools should be part of these devices, that kind of applications and tools which help us every day.

Most of technologies are based on Internet, because Internet access is easy and fast to any device which can access to Internet, any user in the world can access to Internet. 50% of people in the world are Internet user and 2/3 of people in the world are mobile phone user [2].

Nowadays our day to day is easier and more comfortable because of Internet, to inform and to communicate. You can receive new from other part of the world instantly or you can communicate with another person who be thousand kilometres from you.

This concept is useful to monitoring and surveillance, because you can control more than one place and you can see what is happening in that place. Thanks to Internet and networks security nowadays it is possible to install video surveillance installations to control it remotely, installations with movement detection and facial recognition which are some functions you can add to the installation.

This kind of technology is called IP Surveillance [3], which mix close television circuits with IP networks communications. It allows the control can be done physically or remotely through images or audio.

Advance of technology and security allow us to design and implement surveillance systems to control it later remotely. This End of Grade Work look for develop a simple, effective, precise and near tool for the final user, a tool which provides the security the user needs.

Motivation

Nowadays new technologies allow us to develop and implement surveillance systems which can be accessed remotely or physically.

You can manage the surveillance system controlling it remotely, and it facilitates this task for example costs from system decrease. Different detection and recognition tools are some of the advantages that technology provide us to build a surveillance system which be effective and precise.

Physically supervision is done in local, so it is not necessary Internet Access, you only need a local network to can control it remotely, and this is the case when you need Internet to control or to monitoring without being in the same network. This characteristic is a great advantage and advance.

You can use IP cameras which have movement sensors, infrared and different kind of recognition, for example facial recognition. This kind of cameras and their installation don't use to be cheap, their cost vary because of camera characteristics, for example if the camera is able to rebroadcast in streaming, or if the camera is only able to save local to its later visualized, camera resolution, night vision, etc.

Surveillance systems can work similar, camera and sensors are installed, capture protocol is established for example movement detection, facial recognition, time zone, and finally you establish the warning when an intruder is detected (in the case the system consists in it).

Principal motivation of this End Work of Grade is to create a surveillance system which be precise, effective and cheap. This system has to comply basic surveillance system functionalities which have been written before, to its control and its monitoring through Android platforms. So tools like Geany [4] or tools like Android Studio [5] will be used.

Objectives

This End Work of Grade objective is the surveillance system development for Android platform, as in the last parts have been commented.

This End Work of Grade pretends to analyse, design and develop all of that functionalities which can create a precise, effective, cheap and near surveillance system to the user, besides this system have to contribute the security the user require. Surveillance system will be taken charge of rebroadcasting through system cameras images, warning the user in the case of detecting some kind of movement.

Another secondary objectives are register the movement event and provide corresponding event images. Development will be focused to be simple and effective to be the nearest possible to the final user.

State of the Art

In this pulled apart will be focused to describe the environment and context which are nearest to the surveillance system to be developed in this End Work of Grade, besides to describe de system alternatives analysis which are able to implement the system development.

Surveillance

Thanks to surveillance, we can observe multimedia content live (streaming), it can be from our home or it can be from the business. We can observe it from everywhere thanks to remotely control and monitoring through Internet with a computer or with a mobile device.

Surveillance systems are composed by a digital engraver, by a storage support where it saves the video and images, and by the cameras systems which will use the system to take charge of the surveillance. Digital engraver is connected to Internet through a router and it allow us to control and to monitoring images from the system cameras to observe it from everywhere. Control can be done locally or remotely with Internet [6].

Movement Detection

In surveillance systems, movement detection is one of the most important part of the system. You can take charge of this task through 2 options: movement detection via software or movement detection via hardware. Depend on the case, it will be used the first one or the last one or a mix of both [8].

Depend on the case it is probable that the system is focused to separate events and analyse them independently from the system. If the system is developed in movement detection, digital engraver only will attend changes which will be produced in the images, with movement detection via software or with movement detection via hardware, for example with a PIR sensor which allow us to control the movement and the digital engraver.

Facial Recognition

In the last pulled apart has been commented about movement detection aspect from surveillance system which is very important to filter content or to get events which has happened. Nevertheless movement detection is not the only choice that can be implemented in surveillance system, for example facial recognition is another method which can be used in this system.

Same as movement detection, facial recognition is used depend on the final user use, so you can mix both methods in the same system. Facial recognition is considered the surveillance system perfect complement which can improve security system.

Facial recognition serves to recognise a person and to allow access to the person to enter [9]. Software which is taken charge of facial recognition will look for the faces which find through cameras system into a database created before. This database is composed with faces images which can be divide in different sections depend on the surveillance system functionality, for example to divide database in the sections of access control, VIP people or even a offenders section or most wanted people section, for example a section with people who are not allowed to pass into a stadium or into a shop.

Nowadays Solutions

In this pulled apart is going to proceed to describe nowadays principal solutions which exist in market to develop a surveillance system and complying with all system functionalities as well as user functionalities.

It is going to proceed to identify the entity, the organization or the company which offers surveillance services as well as the way they offer it. Here there are some of the most important organizations working on surveillance systems:

- Securitas Direct
- IMSEL Seguridad S.A.
- Altatec

System Analysis

In this pulled apart it is going to do a surveillance system analysis, using Métrica v.3 [23], which consists in get a detailed system specification. This specification have to comply with the users' needs, and it is become in a start point to do the system design later.

System Design

In this pulled apart it is going to do a design which will be taken charge of define the architecture that the system presents. It is going to define technology which will create system structure and it is going to specify with more detail the different components which form the system.

It will start explaining in detail every component which form the system, describing their functions, explaining classes design, physical data design, user interfaces and how files and databases will be presented.

Test Plan

In this pulled apart is going to describe the process which will take charge of checking that all specified functionalities from analysis and design are being fulfilled, considering requirements, cases of use and the architecture.

To get it, it is going to define tests scope and environment to specify later system tests knowing what part of the system and in what environment is going to be tested.

Implementation

In this pulled apart is going to describe architecture implementation and system design which have been detailed before. It is going to proceed to show the most important aspects in development phase after system analysis and design, showing technologies used to implement the project.

Project Management

In this pulled apart is going to proceed to detail the project planning and management which has been carried out during the realization period. It is going to do a comparative between estimated times and real final times, and finally it will be added a project budget estimation.

Regulatory and Ethical Framework

In this pulled apart is going to proceed to describe legal aspect of the project. It is going to detail some standards like technologies licences used in the project. They are the next:

- Android Studio
- Geany
- Python
- MySQL
- VLC

Conclusions and Future Lines of Research

In this pulled apart is going to show the conclusions which has been extracted when this End Grade of Work has finished, proposing possible future lines of research to improve system functionality.

Conclusions

Before beginning this End Grade of Work, it is considered that in the world the surveillance system is exploited, so it should be innovated focusing in a simple, precise, safe and near system, trying to improve the existing systems.

Principal functionality of movement detection systems considered at the beginning of this End Grade of Work has been got great fully, because there are tools, integrated development environments available to get it. The most complicated is to engage these tools to get your proposed functionality objectives.

Surveillance systems are composed by cameras and sensors which have great accuracy but they are expensive. What has been pretended with this End Grade of Work is to do the same functionalities but following a much cheaper way, because using powerful surveillance system cameras is not the same that using a Raspberry Pi, nevertheless in this End Grade of Work is showed that it can get similar results, so it can be an interesting idea for the surveillance besides its adaptation to Android platforms.

In this End of Grade Work, it is pretended to show that you can have in your home or in your establishment a system which warns you when some kind of event happens. It can be cheap and safe and it is not only about capturing, you can configure it on your own, that are some reasons because this idea is interesting and it is much cheaper.

Personal conclusions

Personal conclusions get after doing this End Grade of Work begin when it has been always been something which has been observed so far, like something that never arrives but here I am writing conclusions I get after doing it. I am very happy with the project I have developed, with all things I have learnt during the development.

At the beginning I didn't know if I could develop something like that, but you begin little by little, looking for information, testing tools, asking teachers and the project was taking shape and every time the more I work on this, the more desire I had in this project

As I mentioned, doing this End Grade of Work make me learn more and make me step up my knowledge get 4 years before during the degree, so the sensation you have when you develop an application which works and comply with the functionalities required, make me very happy with all these results got, it is something I saw far away but nowadays it is finished, another finished stage in my life but it is only the beginning.

Future lines of research

Everything can be improved so a software project too. It can be added some kind of improvements to the developed project to improve and optimize implemented functionalities, as well as adding new functionalities. In fact the project is developed to add or to eliminate functionalities in the easy way. Some kind of improvements are the next:

1. It can be added facial recognition function to the system together movement detection. It can be got with another facial images database and implement this functionality in the system.
2. Application user interface can be improved. Most of the interfaces can be improved, it is pretended to use an interface which don't scare the user but it is one of the improvement aspect of the system application.
3. Although system saves movement event with images (because Raspberry has not so much storage to save videos) it can be added the functionality to show these images to the user as well as give to the user the opportunity of visualize event history and watch their contents. It is needed bigger store to develop it but it can be added in the easy way.

